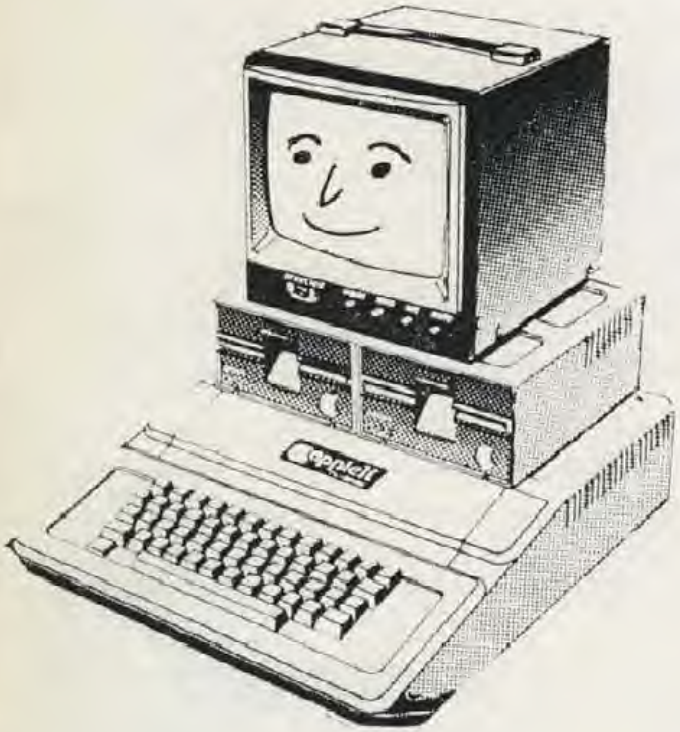
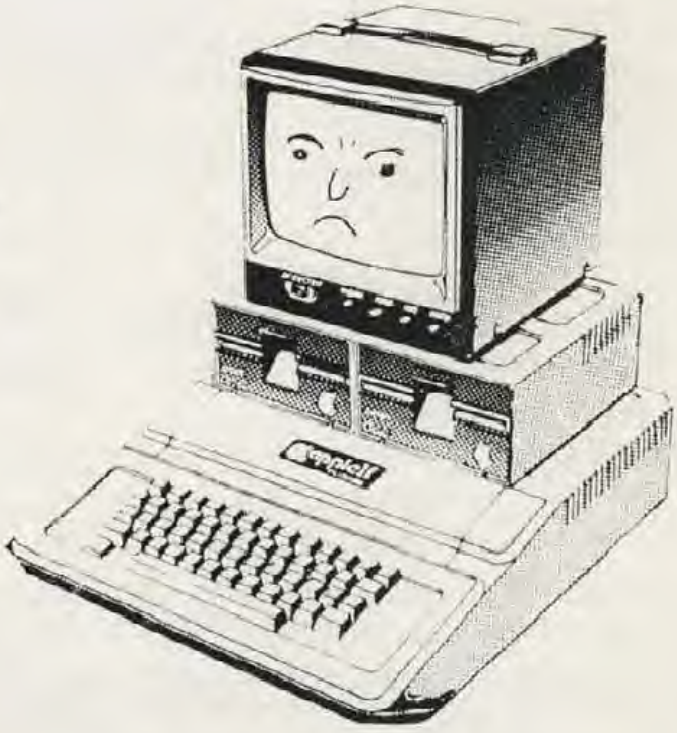
hard core

THE JOURNAL
OF THE
BRITISH APPLE
SYSTEMS
USER GROUP

FEBRUARY 1982 £1 VOLUME 2 No 1





For everything **apple .. ring**

Lux Computer Services Ltd.

108 THE PARADE HIGH STREET WATFORD WD1 2AW
Telephone: Watford (0923) 29513

- Business Software
- Professional Courses
- Pascal
- Recreational Software
- Interfacing
- Cromemco Systems
- Level 1 Service Centre
- Utilities
- Colour Monitors
- Supplies
- Accessories
- Enthusiasm
- Open Tues - Sat

BLUE CHIP SOFTWARE

10 Huson Close London NW3 3JW Tel: 01-722 8827

SUPER TRACE by Ian Trackman £27.50

Super Trace is a high-speed 6502 machine-code utility routine for use in debugging Applesoft programs. As each of your program statements is executed, it is displayed at the bottom of the screen, together with the values of any variable types or functions that you may have pre-selected.

* Programs can be written, loaded and edited without affecting Super Trace
* Does not interfere with DOS
* Can be invoked and disconnected as often as required during program run-time
* Simple to use. Run-time control by single keystrokes
* Optional display of all statements or only statements containing selected variable types or functions
* Variables highlighted in inverse display and their values displayed
* Normal screen display not affected
* 6 run-time speed settings plus instantaneous stop and single-step
* Optional review of previous 10 program statements without destruction of screen display

SUPPORT YOUR LOCAL ADVERTISERS !

This journal is expensive to produce and distribute, but is provided free to members as a benefit of membership.

In order to afford to continue this policy we need advertising to defray these costs.

So DEALERS ! - please advertise with us - the rates are low and we provide a select audience of active Apple users.

And MEMBERS ! - persuade your local dealer to advertise - and whenever you buy from advertisers tell them you "saw it in Hardcore"

hardcore

THE JOURNAL OF
THE BRITISH APPLE SYSTEMS USER GROUP
P.O.BOX 174 WATFORD WD2 6NF

EDITED BY DAVID BOLTON

BRITISH APPLE SYSTEMS USER GROUP

COMMITTEE

FRANK KAY	Chairman
JOHN SHARP	Secretary
DAVID BOLTON	Treasurer
JOHN RODGER	Librarian/ Information Officer
JOHN ROGERS	Software Librarian
TONY WILLIAMS	Membership Secretary

BANKERS

National Westminster Bank Ltd.,
Brompton Square, London SW3 1HH.

MEMBER OF THE
INTERNATIONAL APPLE CORE

COPYRIGHT (C) - The contents of this journal are copyright of the British Apple Systems User Group and / or the respective authors. However permission will normally be granted for non-commercial reproduction by user groups affiliated to the International Apple Core, provided the author and source are properly credited.

The opinions and views expressed are those of the various contributors, and are not necessarily supported by the British Apple Systems User Group.

CONTENTS

4. Editorial
5. Beginners Page — John Sharp
6. Utilities for Programmers - Applesoft Editors Review — Chester Kemp
7. L.A.U.G.H.S.
11. Shapedraw on Tape — Peter Cave
15. Applewriter with a Language Card (Co-Resident Teditor & Printer) — Michael Clark et al
18. Apple Machine Language - Correspondence
21. The Last 0.8 — Cliff Wooton
22. Epson Printer Page — John Sharp
24. Softlib Evaluated - Disk 32 Reviewed — R.D.Purves
26. Education Column — Norah Arnold
27. HELP!!
28. Palsoft Entry Points — Chris Chapman
31. Pippins Page — Vernon Quaintance
33. Microsoft TASC Compiler - a review — Peter Broadbent
35. Catalog Label Printer — John Sharp
39. Readers Letters
42. ITT Hires Pictures — Les Budgen
44. Magic Window Reviews — Tony Williams & Chris Jones

ASSOCIATION OF LONDON COMPUTER CLUBS
Proudly presents
THE
3rd LONDON
COMPUTER
FAIR
APRIL 15th 16th & 17th 1982 10am - 6pm [7pm 16th]
Including: Retail Exhibitors • Hobbyists
Club Stands • Seminars • Workshops
Bring and Buy Sale [17th only]
Admission £1.00
POLYTECHNIC OF NORTH LONDON
THEATRE
⊖ Opposite Holloway Rd. tube station ⊖
The Polytechnic of North London

EDITORIAL

Change of cover style

The change in the cover style has been done at the request of APPLE (UK) Ltd. APPLE do not like anyone modifying the APPLE logo in any way. It is the registered trademark of APPLE INC and, they say, you can either use it as it stands or not at all, or be sued. If you look at the advertisements in the American magazines, the APPLE has virtually disappeared, except as a word in small print. If you don't like the change (we don't), don't tell us - write to Apple and tell them!

Fame at last

The post office generally does a good job coping with the mountains of mail we receive. We have had one or two betting slips for PO BOX 74 in London W1, but the classic was the one addressed to NORWICH - nowhere near Watford really. The full address was the BRITVIC BIG APPLE COMPETITION, PO BOX 15, Norwich, Norfolk !! Is there a BRITVIC APPLE SYSTEMS USER GROUP, we wonder?

Masses of copy - What about the advertisers?

There would seem to be a never - ending source of good articles! Some of them didn't get in yet again (sorry Leo and all you others). We do try to get a balance of material in each issue, which means some contributions get delayed. In addition, if an article is hand or typewritten or on a word processor other than APPLEWRITER, this can lead to delay. We would like to make a larger magazine... but it needs financing by advertising. If all the members who have offered to help could get a quarter page of advertising, we would really have a bumper issue! Can we make a 60 page issue to show at the two fairs in April? We still need articles to match the quality of the ones with which we are starting the new year. Please send them in early, we are busy enough without having to chase everyone. Copy date for the next issue is 15 MARCH.

Apologies for delays

The weather and the sheer mass of mail have been causing some problems over orders and dealing with the mail. We hope we are back on course but the odd error may have crept in so if there is still a problem write in and we will try to rectify the situation. Copying disks has been a headache. One missed delivery because of the bad weather or illness and our schedules are completely thrown out. We tried to have a rest at Christmas but the work only increased because of it. I know we are going to get lots of letters offering to copy disks, but unless you live on our doorstep then it isn't really on. The turnround and the cost is just prohibitive. A better service is on the way; more news next time.

David Bolton

BEGINNERS PAGE

INT IT A PROBLEM

Sorry about the pun, but I hope I drew your attention to a problem when converting real numbers to integer ones in APPLESOFT. Roy Woolhouse has written in with pointing out a problem he has noticed when doing just this. It is something to be very wary when you are programming.

If you are working with APPLESOFT and some of the numbers you use are integers, it is often best to define these as integer variables rather than real ones. An example might be a position on the HI-RES screen, which can only have an integer value. The procedure for assigning an integer variable is

```
10 LET T% = 190
```

rather than

```
20 LET T = 190
```

No doubt some of you will write in and give some better examples. It is always difficult when you are trying to come up with good examples when you are writing, they always occur when you have written the article.

The APPLE then doesn't have to go through the routine to convert it into a coded real variable, which saves time and so makes your program run faster. If you are going to use a number frequently, then you should store it as a variable, because all it has to do is find it rather than process it. This can particularly help to increase the run time of the program if it uses lots of numbers (or strings come to that) constantly.

Sometimes you only want to print the integer part of a number, and then you can do it in two ways:-

```
10 PRINT INT (128 / 34 + .75)
```

or

```
10 H%=INT (128 / 34 + .75) : PRINT H%
```

Now over to Roy's DEMO program to illustrate the problem:-

```
10 REM DEMO OF PROBLEMS WITH 'INT'
20 K1 = 2.6 : K2 = 0.2
30 PRINT "I", "I%", "INT("
40 N = 2 :GOSUB 100
50 N = 12 : GOSUB 100
60 END
100 REM SUBROUTINE USING I, I% and INT
110 I = K1 * N - K2
120 I% = K1 * N - K2
130 PRINT I, I%, INT (K1*N -K2)
140 RETURN
```

A RUN of this program will produce the following printout.

```
I       I%      INT(
5       5       4
31      30      30
```

Now try altering the value of K1 to 2.7 and printing out the result. This time the result is right. So it can be right sometimes but not always. Roy suggested printing I- I% and found the result .999999993 when K1 was 2.6.

This is the clue to why it went wrong, the value of I the machine has come up with is not quite 31 and if it is 30.999999993 then the integer part of this is 30. Roy has an ITT and thought it was due to that machine, but it is due to the way the decimal numbers are converted to binary and back by the microprocessor, and so could happen on any machine, not just the APPLE. The APPLE is good in this respect in fact in that the result is printed as 31 in the first column and not 30.9999999.

So how do we overcome it? Well, quite simply by adding a small number to the result before converting it to the integer value. Thus if we alter lines 120 and 130 to

```
120 I% = K1 * n - K2 + .00005
130 PRINT I, I%, INT (K1 * N - K2 + .00005)
```

all should be well.

There is one point I would like to add which is a hobby horse of mine , although I think there is an exception in this case, because I is the initial letter of Integer. Please don't use I as a variable in a program, so many printers have an I and a 1 looking similar, and it is so difficult to debug a program when you have typed it in wrong. It is just as valid a criticism, if you use 0 and O. Some printers are now printing the 0 with a line through it but there are an awfully large number around that aren't. The worst problem with the 1/I case is with variables such as K1 and KI. Even though it may be clear on close inspection, it usually isn't when your mind is on typing it in.

LOOPING THE LOOP

I had intended to write about loops, before Roy's point came in, so I will make a start and pose a problem. And if there are any problems you have solved or need some help with, I am going to need some ideas for future columns. With more on loops in the next issue, have you any neat solutions, or any particular sticky problems.

The simplest loop uses the dreaded GOTO, for example:-

```
10 INPUT "TYPE IN A NUMBER ";A
20 PRINT "IT'S SQUARE IS "; A*A
30 GOTO 10
```

This is an endless loop, and once the program is running the only way to get out of is to hit CTRL -C. You should put a get-out line by means of a particular test on the input which the user knows how to break the sequence.

The most familiar loop is the FOR-NEXT LOOP. Consider the following:-

```
10 TEXT : HOME
20 PRINT CHR$(7)
30 FOR N = 1 to 1000
40 PRINT "A";
50 NEXT N
60 PRINT CHR$(7)
```

This has a FOR-NEXT LOOP which simply prints the letter A a large number of times. The lines 20 and 60 sound the speaker so that we can time the program.

This program could have been written with another type of loop using GOTO.

```
10 TEXT:HOME
20 PRINT CHR$(7)
30 N=N+1
40 PRINT "A";
50 IF N < 1000 THEN GOTO 30
60 PRINT CHR$(7)
```

Sometimes you are forced to write this way, but in this case it definitely not the way to do it. I measured a time of under 3 seconds for the FOR-NEXt loop and 11 seconds for the GOTO loop, about 4 times as long.

As I said I will develop loops more next time, and so in the meantime with the above illustrations as clues, here is a problem for you to think about. Suppose we wish to print the letter of the alphabet in pairs, so that the letters an equal distance from each end are together, e.g. AZ BY etc. How do you set up loops to do it ?

BASUG?

For those readers who aren't members, it stands for British Apple Systems User Group. Membership is £10 per annum, and there is a one-time joining fee of £2.50. Just send a cheque to our PO Box address - we'll do the rest.

UTILITIES FOR PROGRAMMERS

Notes on four Applesoft editors

by Chester Kemp

If you do much programming with the Apple II, you will eventually decide that you need to supplement its rather basic facilities by getting an editor. There are a number of utilities on the market, and there are always devotees who tell you that theirs is obviously the best - however, I thought it might be useful to take four editors that I have tried, and pass on my experience; hopefully, this may encourage others to pass on their findings.
The four editors I have taken do not cover identical ground, but comparisons may still be of assistance in deciding which one you may wish to buy. The four are CRAE 2.0, Super Editor, ACE, and PLE.

CRAE

publisher: Highlands Computer Services
price: $19.95 + VAT

The Co-Resident-Applesoft-Editor takes up 7K bytes of RAM, but it also offers quite a variety of useful functions within that space. When you wish to use it, you type in &<cr> and get a new cursor (the complementary squared bracket to the usual Applesoft prompt); you are then under CRAE's control.

In the Find command, you can do global searches, so that if you wish to find all instances of variable QQ, or wish to find where all RETURNs are, then all you do is specify it, and the editor then prints every line that shows that string of characters. Further, you can specify it to only check between a particular span of line numbers. An extension to this is when you wish to change QQ to a more meaningful variable name such as TIME, then you use the Change command which searches and replaces all occurrences - it reports every change; if you don't want to see the report, then you would prefer to use the Verify-off command which is identical bar the reporting phase. As you can specify it to change just one line, it can simplify some line-editing considerably. The Modify command is specifically designed to manipulate a single line of BASIC, but it is one flaw in an otherwise excellent package, for if you want to add a bit here, and drop a bit there, it is confusingly presented, and awkward to manage (especially the incompatible insert and delete); I recommend giving this facility a miss, for it is easier to mess the edit up than to correct it.

There is auto-line numbering, for which you can specify the starting point,increment etc., and the Renumbering command is very good. I did a check on the speed of the renumbering and found that for a largish program, the standard DOS 3.3 package was timed to be 22.7 secs, for the DOS 3.3 Toolkit it was 8.2secs, but for CRAE it came out at 4.6secs!! For those who have not used RENUMBER, it not only resequences the line numbers, but it also changes any GOTOs and GOSUBs in accordance with the new sequence. Two remaining line manipulations are available: Quote (which takes a straight copy of some lines and places them without modification into another specified place) and Append (which allows you to merge a piece of coding held on disk, to a program already held in RAM).

where is a List command, which has slightly different parameters to the Applesoft command, but presents your listings without any of the unneccesary spaces, whilst still occupying the full 40 columns of your screen. This can be stopped and started by a single keystroke such as the space-bar.

Finally we get to the area of looking beyond Applesoft. The Dump facility, gives for specified start and stop addresses (in hex), the hexadecimal contents, and their ASCII equivalents, eight to a line. The ASCII representation gives both inverse and flashing representations, but a number of codes are not translated except for a full stop to mark its place.The Quick and Dirty section allows you to:- enter monitor commands without leaving the editor; make decimal/hex conversions; give the number of free bytes between the end of your program and HIMEM; and also the decimal length and address of the last BLOADed program (in this, it is not so convenient, as you still have to convert them to hex to subsequently use the Dump command).

The documentation is very good, and apart from reservations on the search technique (mentioned below), the package is a joy to use. It can also be co-resident with the PLE package.

I am given to understand that CRAE has also been known as PGE (Program Global Editor), authored by C A Greathouse & Garry Reinhardt available from Call A.P.P.L.E.

Super Editor

author: Ian Trackman
publisher: Blue Chip Software
price: $27.50

This package by Ian Trackman, does not have many facilities, but what it has, it does speedily and elegantly.

There are essentially two 'commands', one for global searches, and the other for search and replace (either totally or selectively). He has made the distinction of separating what is literal text (such as data statements and remarks), and tokenised commands and variables; the result is that we have a particularly speedy utility. Looking back at CRAE, it is considerably slower, for it converts all tokens into characters for its own search. The benefits of CRAE's approach are perhaps in a theoretical flexibility, but in practical terms there are no real benefits. The only extra consideration for Super Editor is that when you use a BASIC command, it must be complete, e.g. not TAB, but TAB(.
To access the editor, it is always necessary to commence with &" or &$" (for respectively commands/variables or text). The elegance I mentioned is the highlighting of each successful find, whether for replacement or not, by inverse video of the relevant characters; this is a very simple but extremely helpful device. On the negative side, I am irked by the repitition of a line when the object of search occurs more than once on that line. If this bug could be taken out it would be pleasant (although it may be awkward when asking whether a particular instance needed replacement or not), but as it stands it is still particularly pleasing.

I personally find this a little over-priced, but Ian Trackman tells me that the price will not change even if the value of money decreases! This editor (occupying only 1.5K bytes of RAM) will co-exist with either ACE or PLE, and may be felt to be an ideally complementary.

L.A.U.G.H.S
(LEICESTER APPLE USERS GROUP FOR HELP & SUPPORT)

The Leicester group is going from strength to strength. We have at present grown to 40 members. In fact we have grown to such proportions that a change of venue will be on the agenda in the very near future, so keep an eye on this column for future news. In the meantime we still meet at the premises of the LEICESTER COMPUTER CENTRE, 67 REGENT ST, LEICESTER on the first Wednesday in the month at 7.30 pm.
Should you wish to contact us between meetings, please ring either the secretary Mrs Hazel Brown on Leicester 875253, or Mike Preston on Rearsby 647.
The next meeting will be centred on VISICALC and associated programs, and will be followed by a discussion night, and some special lectures on simple programming techniques.
We are trying to cater for every taste, so why not come along and join in the fun at the next meeting.
MIKE PRESTON
CHAIRMAN

ACE

authors: Jeff Baron & Roger Clayton
publisher: Southwestern Data Systems
price: $19.95 + VAT

The Applesoft Command Editor is "an integrated package of powerful subroutines". It has two levels of commands, firstly, single letter macros (such as C for CATALOG), and secondly macros which require the prefix of the Control key (which can be defined by the user at any time). The package occupies approximately 6.5K bytes, and resides just below DOS.

Let us first look to the line editor (accessed by Ctrl E then the line number). The line is put on the screen with the cursor upon the first character of the line number; you can now jump forward and backward with good flexibility (using some Ctrl letters which are uniquely used in Edit mode); if control characters appear in the line, they are printed in inverse video. To delete and add characters within a line, the use of, respectively, Ctrl-D and Ctrl-I, are easy to handle. Upon getting the line amended, whether the cursor is at the beginning, middle or end of the line, the Return key transmits the whole line; to abort the edit type Ctrl-X, and the line is left untouched. This is in fact the source of a bug, for if you abort the edit on column 38, then the Applesoft cursor also falls in that column until another Return (a small point, which causes no practical bother). Another bug is on the Ctrl-C, which takes you to the character after the next colon - but in fact if the next statement has an = in it, it places you to the next character after that instead, and passes over its intended stopping place; this I found to be irritating.

Besides the edit mode, you can use 19 letters of the alphabet to define your own macros. Simply type Ctrl-D followed by the letter, and give your requirement. Each macro can be up to 7 characters long, so that if you are writing a program requiring many colour changes, then you define a macro with COLOR=; you can do this at any stage without any having to load in another program to create this. I initially found it frustrating as I wished to use them for setting up the printer, but you can get round this by having a series of control keys to set up e.g. the print font and number of characters per line.

The Abbreviated commands are quite powerful, for they cover a number of useful utilities. The List allows single key stop/start control. The F gives the free sectors left on the disk. The M gives the decimal values of HIMEM, LOMEM and FRE(0).

The B gives the hex address and length of the last BLOADed program. There is a facility to do hex/decimal conversion by using the letter X, and it recognises which way to convert, because the $ sign precedes the hex (to be preferred to CRAE's separate command characters, which are not always memorable). The Dump facility only requires the start address (in decimal or hex), and then it gives a dump similar to the CRAE, and similarly single-key controlled. The differences in the ASCII representations are interesting though, for CRAE gives inverse and flashing, whilst ACE just gives them normally - ACE does not translate the open square bracket, back-slash or the under-line.

On the home straight, we have the ability to include the standard Master 3.3 RENUMBER (having first made the corrections mentioned in Hard Core issue 5, page 45!),as well as executing monitor commands directly. The remaining goodies are very useful: XR and V. XR followed by a variable, lists all line numbers that it appears - naturally it can be done (complete with the whole text line) by Super Editor and less conveniently by CRAE - but it is simple and efficient. The V command gives all variables used during the running of a program, complete with the order used and the current value held. This can be of use in de-bugging as well as optimisation (where oft-used variables require early definition.

The documentation is good, and the package handles well. A COPY utilitity is included, which allows back-up copies to be made, but it only allows three copies to be taken.

PLE

author: Neil Konzen
publisher: Synergistic Software
price: $22.00

The Program Line Editor has an edit mode as well as an ability to create powerful macros. The edit mode is entered identically to ACE, but the cursor now commences by pointing to the first character of the line (after the line number). It manipulates the line in various ways, but it differs in ability from ACE in that it can change for example, the comments from upper to lower case, which can be more clearly read from hard copy.

The important strength of PLE is in its macro capability using the ESCape key. There is a total of 333 bytes available for defining macros, although the default values do not initially leave a lot of room for adding your own. As I work in Applesoft, I simplified macros which calculated addresses (where numbers over 32K have 64K subtracted from them to keep the range within +/- 32K) to a range of 0 to 65535; this saves quite a few bytes. My own requirements were to have printer type styles called, so that I could change from 10cpi to 16.7cpi, and also to proportional print etc etc. The seven characters in ACE were inappropriate, but PLE was able to cope without difficulty. Maybe Ian Trackman will get the itch to find a way to double this 333 to 666 (the number of the Great Beast!?), for this would be one improvement I would be happy to see, but can still live without if I have to.

Default macros include ESC1 and ESC2 for catalogs of drive 1 or 2; ESC0 is HOME; ESC: gets you into the monitor; ESCW gives HIMEM and LOMEM. In ACE, you could change your macros as you went along, but in PLE you have to call another program which will display, edit and create macros. Naturally, you could store several different sets of macros for differing purposes. This package resides high in memory between DOS and it's I/O buffers, and only takes 600 bytes.

Final Comments

There is often a need to have a hard copy of your dumps, and sometimes edits, and whilst I had no problem using Super Editor and CRAE, in using the create macro phase of PLE the printer was disconnected. If you want to print when using ACE, then you have a real problem, because activating the printer disconnects ACE's facilities, requiring CALL0 to reconnect; to get round this inconvenience, I find that Mike Glover's Screendump (in Windfall no.4) very useful.

There is no outright winner, because they all do different things. Both ACE and PLE can be co-resident with Super Editor, and in doing so complement each other pretty well; however, the combination of PLE and CRAE is pretty potent as well. My own preference when doing a lot of programming would probably by ACE with Super Editor, but I am still very attached to CRAE. On the point of copying, both PLE and CRAE are un-protected, but Highlands Software states "We believe in low cost software with profits coming from volume sales. We also believe in a buyer's right to easily make backup copies. However, unless the problem of 'free copies' is brought under control, those of us who provide low cost software will be forced out of business, leaving those who sell low volume, high cost software! Please do not abuse our trust. Thank you." I don't know whether they will be dis appointed, but I hope that utility software does not increase in price, for I feel that utilities (not arcade games) are the life-blood of computing.

It is inevitable that there will be points I have overlooked, or that you may disagree on the amount of importance I place upon a particular facility, so that I hope that others will write and expand the forum.

Wida Software

Specialists in Educational Software for Schools

APFELDEUTSCH.....£ 120.00
CAL course in German; Beginners to O'Level and beyond. Textbook: Workbook: 6 Language Lab cassettes: 9 Apple disk of teaching and testing routines.
Apple only. (20% discount for schools)

TEACHER'S TOOLKIT.....(Apple disk) £ 20.00
(Pet cassette) £ 15.00
Starter pack for building up library of tests. No knowledge of computing needed. Suite of five programmes; Tester, Testmaker, Editor, Multiple Choice specimen, Directory of tests

LERARENPAKKET.....(Apple disk) £ 25.00
Teacher's Toolkit as above but with on-screen instructions in Dutch

GERMAN ROUTINES.....(Apple disk) £ 15.00
(Pet cassette) £ 10.00
Stand alone teaching and testing routines - article and adjective endings, word order, pronouns, etc. Send s.a.e for details

PEDAGOG AIDS.....(10 Apple disks) £ 60.00
Life raft for teacher thrown at the deep end of Computer Studies. 20 Apple programs taking you to O'Level.

SHAPE MANAGER.....(Apple kit) £ 59.65
The easy way to build up shapes and use them in your program. Does for shapes what a word processor does for words. From Sinta Software

TYPE-RIGHT.....(Apple kit) £ 46.50
Lower case word processing with Apple Writer. Full shift key operation. Plug-in fitting with disable switch. Manual, fitting instructions, software on disk.

APPLE PILOT.....(2 disks) £76.00
The ultimate language for teachers; mix sound, graphics and text on-screen in your teaching programmes. And much else. Disks and manuals

ARISTOTLE'S APPLE..... (Apple disk)£ 20.00
Authoring system; easy to use multiple choice matching tests. Includes alternative answers.

LUDINSKI'S LOGS..... (Apple disk)£ 20.00
O'Level revision of logs problems. Teaching, testing and scoring. Correct solution rewarded by 25 seconds of arcade game.

COMPUTERS FOR KIDS.....(book) £ 4.00
A Basic programming manual written just for kids. Special section for teachers and parents. by Sally Greenwood Larsen Apple II plus edition

TIMETABLING	Let Keith Johnson - England's
TIMETABLING	foremost authority on time-
TIMETABLING	tabling guide you through
TIMETABLING	next August's trauma!
TIMETABLING	Book with CBM listings £8.95
TIMETABLING	Apple software on disk £30.00
TIMETABLING	Software on cassette for
TIMETABLING	380Z, TRS80, CBM £30.00

All prices inclusive of VAT postage & packing free

Contact Wida Software
The Red House
2 Nicholas Gardens
London W5 5HY
Tel (01) 567 6941
or (062 882) 5206

VISA

BOOKS FROM MICROSOURCE

NEW! NEW!

GRAPHIC SOFTWARE FOR MICRO COMPUTERS

by B. J. Korites

Over 100 LISTINGS IN BASIC THEORY from simple Algebra to Matrix theory
Suitable for BEGINNERS/SCHOOLS, COLLEGES
Starts from plotting points to sophisticated 3D Hidden Line Removal, perspective and shading
WRITTEN FOR APPLE II but easily adapted for RESEARCH MACHINES 380Z

BOOK	£15.95 inc P&P
DISK of BASIC LISTINGS for the APPLE	£15.95 inc P&P
2 DISKS of the same programs in MACHINE CODE	£16.95 inc P&P

BENEATH APPLE DOS.

By Don Worth & Peter Lechner. A Technical Gold Mine of the secrets of DOS (3.3 and 3.2). Explanation of how the disk is formatted and accessed as well as detailed listings of the routines. £11.95 inc P&P

ALL ABOUT APPLESOFT

The American User Group CALL A.P.P.L.E have reprinted some of their own articles together with some from other sources, and then had specially written the ones that were missing to give an excellent compilation of the ins and outs of APPLESOFT in one volume. Contents include -

—APPLESOFT FROM BOTTOM TO TOP — a guided tour through APPLESOFT both a description and hundreds of listings and programs and subroutines to help you get more from your APPLE
—MAKING BASIC BEHAVE — a structured approach to BASIC with numerous listings of ideas and subroutines for formatting, file handling etc
—INTERNAL ENTRY POINTS — a guide to the various machine code routines in APPLESOFT that you could use in your own programs. Not only where they are but how to use them. A companion to "WHAT'S WHERE IN THE APPLE"
—NOTES ON HI-RES GRAPHICS ROUTINES — a specific article on the use of the HI-RES GRAPHICS routines to help you to write machine code graphics
—AMPLIFYING APPLESOFT — adding extensions to the language as well as methods to get the best mileage out of APPLESOFT, again with lots of listings
—PLUS lots more, print using, splitting programs, comparing programs etc etc
—IN SHORT a must for all serious APPLESOFT PROGRAMMERS £9.95

WHAT'S WHERE IN THE APPLE

By William Luebbert

The Atlas of Peeks, Pokes and Calls — listed numerically and alphabetically
Suitable for APPLESOFT INTEGER and MACHINE CODE programmers, covers both BASICS, Monitor and DOS.
128 pages £9.95

MICRO on the APPLE

A series of Volumes of Articles on the Apple from MICRO 6502 magazine. No need to type the listings, all programs on disk (included with each volume) contains:- Utilities, Games, Graphics Education and Hardware articles.

VOL 1	£18.95
VOL 2	£18.95
VOL 3	Coming Soon

NEW! NEW!

RAMPLUS BOARD (NOT ROMPLUS!) from Mountain Computer expand your APPLE to 80K with two banks of 16K RAM, hardware or software switchable. Provides all APPLE LANGUAGE CARD functions plus an extra 16K of RAM. Lower power drain than a normal RAM card £120

THE ENHANCER II — allows words or phrases to be defined at the touch of a key! Allows you to continue typing whilst the APPLE is busy doing other things, create a new keyboard layout with a different plug in chip, enhances the use of the APPLE for WORD PROCESSING, SECURITY systems etc etc. Gives upper/lower case on revision 7 or later APPLEs £100

SPECIAL SOFTWARE

As well as being able to get almost any existing APPLE SOFTWARE or peripheral quickly, we have a team of writers producing the software that we as users ourselves wish was available. This month's new offering is -

BUSINESS BASIC

Now you can have APPLESOFT with the features APPLE left out
—Formatted output
—Programmable TAB fields
—Bi-Directional Scrolling
—INPUT ,; in strings
—DISK store/recall commands giving 3 fold speed increase and up to 3 fold space reduction
—Direct access to disk sectors
—'HELP' command for debugging
—'SCREEN' or 'PAGE' output
—INCLUDES the MINI ASSEMBLER
REQUIRES A 16K RAM card for use £28

ALL PRICES INCLUDE VAT UNLESS SPECIFIC MENTION IS MADE ALL PRICES INCLUDE P&P

MICROSOURCE
Tel: Park Street (0727) 72917

1 Branch Road,
Park Street
St. Albans.

Accutrack Disks . . .

Because data reliability is the important difference in disk construction.

Anatomy of a disk.

Flexible disks are simple information storage devices consisting of a magnetic disk enclosed in a semi-stiff protective jacket. The disk rotates within the jacket while magnetic recording heads on your data or word processing systems "read" or "write" information on the disk's magnetic surface. Since disk operation is simple, it's relatively easy to make one that works. But building in reliability is something else again. It takes specialized technology to build disks that operate flawlessly over an extended period of time.

What counts in disk construction.

Key design objectives for a disk are listed below. How well a disk measures up to these objectives relates directly to the throughput, accuracy and overall costs for your data or word processing system. No disk measures up better than Accutrack.

- The magnetic coating must be precisely formulated and uniformly applied. Imperfections as small as five millionths of an inch cause signal dropouts, data checks and wasted processing time as well as errors.
- The disk surface must be absolutely clean, totally flat and permanently lubricated to prevent excessive head wear with subsequent signal degradation and eventual loss of information. (This is the most critical objective and the one that's most often compromised since poor operating results take a while to show up. It's also the area that most effects the long term reliability of your data.)
- The disk must be free to rotate within its jacket without internal drag to avoid further data checks, excessive processing times and errors.
- The jacket must protect the disk from external contamination and damage. It should also remove microscopic particles of debris from the disk surface before they can damage the disk.

Why you'll never find the best disk bargain in the bargain basement.

While there's little apparent difference between other disks and Accutrak, the performance differences can be substantial. Simply stated, an Accutrack disk is premium priced. But the protection it gives your information; the reliability it provides to your operations, and its substantially longer life make it the best disk buy. After all, the real cost of your operations is constructing and processing the data stored on the disk - not the disk itself. It doesn't make sense to trust that data to anything but the best disk. Accutrack

BASUG BULK BUYING — 50% off PRICE FOR 10s

£17:50 incl FREE LIBRARY CASE !

(ADD 50p POSTAGE / PACKING)

BRITISH APPLE SYSTEMS USER GROUP

P.O. Box 174, Watford WD2 6NF.

SHAPEDRAW ON TAPE

by Peter Cave

This monograph is intended mainly for Apple users who must rely on tape, although disk owners may find items of interest. It comes in three parts:

1. A description and listing of a shape table compiler program.
2. Some notes on hi-res shapes, particularly a full alphanumeric character set.
3. Joining a shape table to a BASIC program.

1. A shape table compiler

The description given in the Applesoft handbook on the construction of shape tables is clear and readable; whether anybody wants to take the trouble to use the methods having read it is another story. To use this facility to compensate for the lack of lower case letters is enough to make the experienced programmer reach for the Valium. This explains the proliferation of programs designed to ease the difficulties so that writing a shape table can be almost as easy as writing a letter. The use of these programs can encourage a 'black box' attitude to the process, but with a little care and effort a very versatile table can be constructed.

SHAPEDRAW, the program offered here has distinct advantages: it is very user friendly, is not disk dependent, and will cost you nothing. With forethought it can be made to work almost as well as commercially available disk-only systems. Disk drive owners can use it profitably with the suite of programs on the BASUG introductory disk.

The original concept was devised by Malcolm Banthorpe and appeared in Practical Computing for September 1980; bugs were ironed out in letters in the issue for the following December from outraged Dirk Rietveld and Neil Lomas. My contribution has been to make the program more memory efficient, supply more information to the user and to transfer the design grid from the lo-res screen which inevitably gave a distorted image, to the hi-res.. The magnified grid for the design of each shape was retained for clarity and to match graph paper layouts. This means that the design grid and shapes drawn on it are four times larger than when SCALE=1.

The program begins by asking for the horizontal and vertical dimensions of the grid to be employed; the same grid will be used throughout the run of the program. The origin of each shape is at the bottom left hand corner although this can be changed by modifying line 1070

Any shape table can hold up to 255 shapes. The index to the shapes precedes the definitions; in order to effectively allocate memory to the index the user is asked to enter the number of shapes to be contained in the proposed table.

Allowing space for variables' use, the program establishes the free space around the HGR1 screen and asks for the decimal starting address. Deletion of REMarks, reducing the variable names to two characters and rewriting the introductory passages in 4100 to 4180 to suit your own working will relase further memoery.

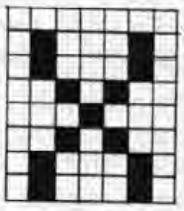
When the operational parameters have been entered, the grid and plotting cursor are drawn, the number of the current shape and the bytes used are given along with the commands. PLOT will switch the cursor to a white block and MOVE will switch it back to a dot. The cursor can be moved with the vector commands. Any movement following two upward moves will automatically cancel the shape as the construction cannot be accommodated at machine level. SEE will draw your shape to prescribed SACLE an d ROTation. X will cancel the attempt if you have made a mistake; E will enter it intro the shape table and prepare the screen for the next image.

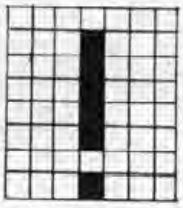
When the table is complete the length, starting and ending addresses are given in both decimal and hexadecimal, with details for saving to tape or disk. It is particularly important to make a written record of this data, at least of the table length (TL), if further work on the table is anticipated.

2.Some notes on shape table construction

While my main interest has been in the production of letters and other characters, it will be apparent that the remarks made here can be applied to any shape table.

On the Apple each ROM character is designed to fit inside a 5 x 7 grid which itself is contained inside a border area; thus X and ! both occupy the same amount on screen.





Two elementary division sums, 280/7 and 192/8 reveal that the full hi-res screen can be made to look like the 40 x 24 text screen and it is this option that most designers of lower case characters adopt: lower case letters are forced into an identical grid and are DRAWN on the hi-res screen so that they coincide exactly with the upper case letters on the text screen. The two screens are then driven by the character output routine, although generally only one screen is visible.

If you are using your computer as a text processor then you may be fairly satisfied with the slightly distorted font that this method produces. But as the printers used in the production of this magazine show us, not all characters require the same width and printing has a greater aesthetic appeal when proportionally spaced. If your printed output to the screen is to be limited to a page or two of instructions or a few words of text then your own alphanumeric table would be useful.

When a shape table is compiled a series of orthogonal vectors are coded to form a miniature program or map that the computer can follow from start to finish. When DRAWing a shape it is usually necessary to give screen coordinates for the start of the shape and the computer will then p'pace' out the map from the start point to the end of the sequence.

If is asssumed that a line of text is to be written at the same level on the screeen, then it would be sensible to start each character in the table from the same point so that a constant variable may be used for the vertical parameter. A program to write a proportionally spaced three letter word would like this:

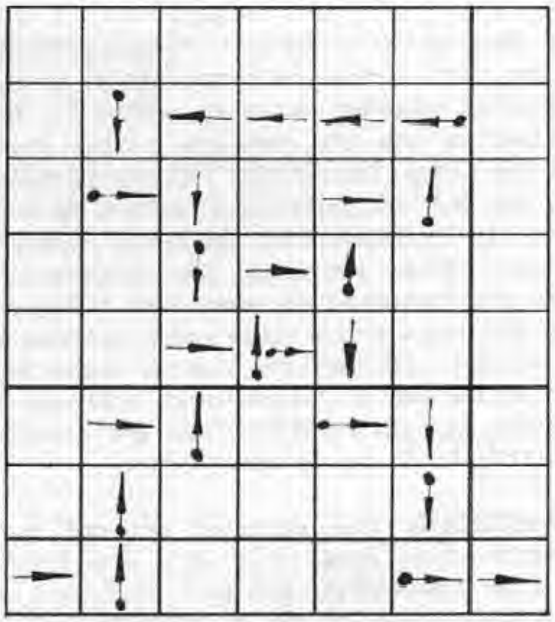
```
10 H = 20:V= 10
20 DRAW 1 AT H,V: H = H + 8
30 DRAW 2 AT H,V: H = H + 6
40 DRAW 3 AT H,V
```

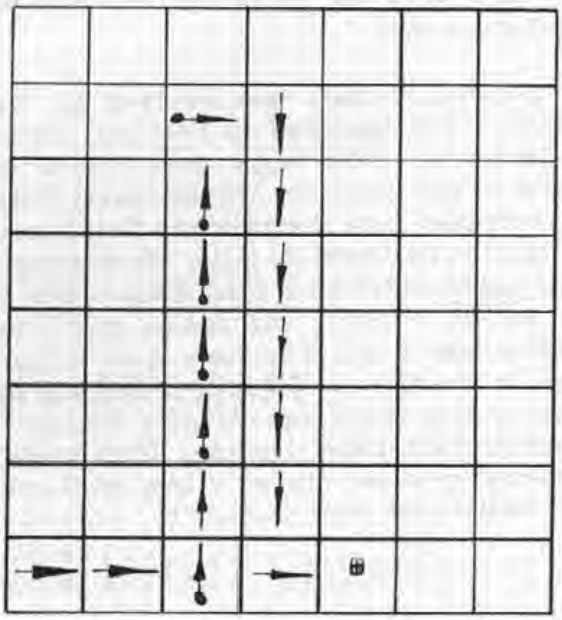
This would be tedious for a page of text so a further attribute of the Apple can be exploited. Having made reference to the zero page stores for the coordinates of the first character, the computer will DRAW a subsequent shape from the point at which it left off the first if no further coordinates are defined. Consequently if all shapes in a table are designed with the same entry point to the grid and concluded with an opposite exit point and space is allowed for gaps between characters, any shapes DRAWN after the first will appear proportionally spaced. The program above can be reduced to:

```
10 H = 20: V=10
20 DRAW 1 AT H,V: DRAW 2:DRAW 3
```

A line of text can restricted to a definition of the first shape and its coordinates, a loop and data statements. A further method will be outlined later. Extreme care must be exercised in the choice of vectors both to ensure that entry and exit points are correct and to conserve memory. It is good practice never to finish a shape with upward moves only as these will be compiled to null bytes, ending with the shape earlier than you had intended. It is a good idea to draw each shape out on graph paper and describe the path through before trying to code it even if you are using a compiling program.

Following the convention described by Apple, our two characters would appear on paper in this form:





@ is the last vector plotted and will be the starting point for the next shape XDRAW.
You will not be able to SCALE up the majority of alphanumeric characters as the expansion of each shape is contingent upon the vectors used. On a larger scale the X looks like a pair of crossed staircases. If you need a larger font you will have to use a larger shape from the outset. Other shapes are easier to manage. As drawn the ! will show a projection to the right if made larger. This can be avoided if the final plot and move comes back down the shape, overwriting the previous plots. Subsequent vectors can be moved along the shape without harm and then off to the exit point. All regular shapes that are to be enlarged can benefit from this procedure although it is not always necessary to aim for the same exit point if consecutive drawing is not the objective.
Similarly, closed shapes such as squares will only be complete if the final plot overlaps an earlier. If this precaution is not taken then the shape may not be complete when SCALEd up. SHAPEDRAW will allow you to experiment until you are satisfied.

The way in which you select shapes for your table is entirely up to you. I devised a general purpose table to be used whenever I need alphanumerics. My first 64 shapes match the ASCII characters 32 to 95 with the exception that the alphabet section consists of lower case characters. If I need to use this table with an interpeting routine to write directly to the screen from the keyboard (using the ESC key to shift) I need the upper case letters to be separated from their lower case equivalents by 64 places. I could have duplicated the pre-alpha series or, as I can access those shapes already, I have devised 33 other shapes to take their places. So now I have a full character set with the addition of other shapes which may be useful. Instead of duplicating the enabling routines used above I can now use a third method, in the form of a formula, which can be used to write a word or phrase:

```
10 DRAW 98 AT 10,10
20 FOR I = 1 TO 29
30 DRAW ASC(MID$("N APPLE IS A
MAN'S BEST FRIEND",I,1))+J
40 NEXT I
```

J = (-31) for lower case letters and J = 33 for upper case. By varying the value of J and carefully selecting letters for the source string I can DRAW any of the 128 shapes.
The 33 new shapes in the middle of the table create a buffer zone. If I wish to use the general purpose table in a program but need shapes that are not already there, they can be placed in the buffer zone with the loss of unwanted shapes.

If you are using SHAPEDRAW each new shape can be created one at a time. Having designed the shape, enter the monitor list from the start address, ignoring the disassembled mnemonics. The shape definiton begins at the fifth byte from the beginning of the shape table and extends to the first null byte. In addition to the data recorded when the full table was compiled there further notes should be made.

$SL = length of table - 4 (shape length)
$SS = start of table + 4 (shape start address)

$SE = $SL + $SE (shape end)

Then save the data either by copying it from the screen by hand or printer or,m more tricky, saving the definition on tape or disk. To save just the shape definition and not the whole table tape users should type:

SS.SE W

and disk users:

BSAVE shapename, ASS,LSL

Use SHLOAD or SHAPE-PROG from the BASUG disk to load your major table. When you have used this program execute a PR#6 or strange things may happen. As the map on page 95 of the Applesoft manual shows, a complete shape table has three parts:
1. Record of the number of shapes in the table.
2. Index to the shapes with the addresses relative to the start.
3. The shape definitions.

With S as the number of the shape to be replaced or changed there are three ways of finding it in memory. All require extensive use of the monitor to be used effectively and loading a DEC/HEX-HEX/DEC routine may be useful if you are unfamiliar with working with BASE 16. Remember that simple arithmetic operations can be undertaken by the monitor. Brief and incomplete outlines will be given of the first two methods and the full processes are readily deduced. Both require immediate entry to the monitor. Find the start of the shape table by typing E8.E9; hi and lo bytes are, like all addresses in monitor, reversed.
1. Emulate the computer by counting S+! pairs of bytes from the start of the table. Add the number found in the last two bytes of the series to the addresss of the start of the table. This will give you the location of the start of shape S.
2. The second method involves examining the shape definitions themselves. As each shape ends with a null byte, finding the address of the byte following a 00 will reveal the start of a shape.

3. The third and simplest method is to let the computer tell you where a shape definition starts. While in BASIC type DRAW S-1. Enter the monitor and ask for the contents of 1A.1B. These will give you the address of the final byte of the shape preceding S. If you list from that address you will find that the first non-zero byte will be the first byte of shape S.
Whichever way the address is found the first byte will be called NS. The new shape can be loaded by typing in the data from the keyboard if it was copied by hand or printer. It can be entered from disk straight into this area using BLOAD shapename, A$ NS. Tape users will have to be a little more careful: find the new shape end (NSE) by adding NS to SL and type NS.NSE R before running the tape.
The new shape may have overwritten more than one of the former shapes. Test for this before loading another new shape by returning to BASIC, turning to a hi-res page and DRAWing S, S-1, S +1.
To save the modified program tape users will have to reenter the monitor. Store the length of the table by typing 0:TL(lo) TL(hi). Find the start of the table by calling on E8.E9 and the end (TE) by adding TS to TL. The new table can then be saved by typing 0.1W TS.TEW. Disk users must also find the starting address before typing BSAVE tablename, ATS, LTL.

3. Adding a shape table to a BASIC program

The impecunious amongst us do not have the benefit, like those with DOS, of being able to load a shape table from within a program and must resort to loading two separate routines. It is possible to convert the machine code program into a series of POKEs, but even entering these as loops and DATA the program will be considerably lengthened and delay experienced while the conversion takes place. It is more efficient to make the table part of the program into its unconverted form.
The BASIC programs appears in memory in numerical form and sections after deleting, adding or changing a line are moved and the end of program pointers and variable space markers are adjusted accordingly. If POKE 232,TS(lo): POKE 233,TS(hi) is added to the program to tell the computer where the shape table is, then the table can be appended to the program and the end of program pointer changed to indicate the end of the shape table. This is work provided that:

1. The program itself or its variable space does not extend across the boundary of the HGR page that you are using.
2. That no line is added or subtracted.
3. That the program is not then loaded into an unusual place in memory.

But a fully relocatable and editable program can be made to work. Before trying to join a shape table to a program add these lines:

```
START = PEEK(175) + PEEK (176)*256-TL
HI= INT(START/256):LO - INT (START -256*HI)
POKE 232,LO:POKE 233,HI
```

Tape Users Only
a)SHLOAD your shape table
b)enter the monitor and find the end of the program (EP) by typing AF.B0
c) find the start of the shape table (TS) by typing E(.E9
d) find the end of the shape table (TE) by adding TS and TL
e) move the shape table to the end of the program by typing EP<TS.TE M
f) GOTO 3

Disk Users Only
a) enter the monitor and find the end of program (EP) by calling on AF.80
b) BLOAD tablename, A$ EP

All Users
a) the new end of program (NEP) will by EP + TL
b) reset end of program pointer by typing AF:NEP(lo) NEP (hi)
d)SAVE the revised program

WARNING

DO NOT USE APPLESOFT RENUMBER ON THE REVISED PROGRAM. THIS WILL RESET THE END OF PROGRAM POINTER TO THE END OBASIC LINES AND THE SHAPE TABLE WILL BE LOST


```
 GGG    TTTTT  RRRR    -A    N   N
G   G     T    R   R   A A   N   N
G         T    R   R  A   A  NN  N
G         T    RRRR   A   A  N N N
G  GG     T    R A    AAAAA  N  NN
G   G     T    R  R   A   A  N   N
 GGGG     T    R   R  A   A  N   N
```

GTRAN is a graph plotting package written in BUSINESS BASIC for the EPSON MX 100 and MX 82 printers. It plots graphs on six graphics screens to give a resolution on the printer of 560 for the X axis and 576 for the Y axis (excluding scales and labels).

The package also contains a numeric screen editor with macro facility and the ability to generate X and/or Y data. A scaling program that allows upto 10 plots from upto 10 data files and 1000 points in total. The scaling program can also apply a set of functions to the data. A program is also included to apply a moving average or linear regression to the data.

The price of GTRAN is £28 or £50 if bought with BUSINESS BASIC. See MICROSOURCE for details.

BUSINESS BASIC

BUSINESS BASIC is upwardly compatible with APPLESOFT except for tape commands and LET. It runs on a RAM card and is 16K in size instead of the usual 12K for APPLESOFT.

BUSINESS BASIC offers you the following features:

Programmable tab fields with a CONTROL I or comma giving a tab.

Tabs can be expanded into spaces when sent to peripherals or left as CONTROL Is'.

SCREEN or PAGE output.

Formatted output of numbers defaulting to normal APPLESOFT.

Bidirectional scrolling with UP and DOWN commands.

Input anything to strings using a LIN command.

Disk STORE and RELALL commands. Direct access to disk sectors and disk formatting.

MID$/LEFT$/RIGHT$ - string.

"!" is a help command for debugging.

Listing is full width and the mini assembler is included.

The price is £28. See MICROSOURCE FOR DETAILS.

APPLEWRITER WITH RELOCATED DOS

By Michael Clark

When I use Applewriter I find that I am constantly going backwards and forwards between the editor and printer files to make corrections and additions, which has meant reloading the other program each time the transition is made. But I have now speeded this up by making use of the language card to relocate DOS. The July/August 1981 issue of CALL A.P.P.L.E. contains a program, DOS MOVER, to relocate DOS 3.3 and I have exploited the extra space it makes available by storing copies of TEDITOR and PRINTER in it. Rather than get involved with the problems of relocating one of these programs and running it from a new address I use the ampersand to call a short machine language program to move the required program from high memory to run at its usual location.

Here in detail is how to do it:-

1. Initialise a new disk with DOS 3.3 patched at $9E42 [replace 06 by 34]:

INIT DOS MOVER

and then delete DOS MOVER from the catalogue.

2. Transfer DOS MOVER and the Applewriter files TEDITOR, PRINTER, PRINT CONSTANTS and TRANSFER FILE to the new disk.

3. UNLOCK TEDITOR AND PRINTER. BLOAD TEDITOR, enter the monitor and patch as follows:

```
*1118:DD
*111D:DD
```

Now BSAVE TEDITOR,A$803,L$1040. BLOAD PRINTER, patch it from the monitor:

```
*A59:DD
*A5E:DD
```

and BSAVE PRINTER,A$803,L$1040. Both files can then be locked.

4. Assemble HELLO - listed below - and transfer it to the new disk.

NOTES

(i) The patch at $9E42 makes DOS run a binary file as the greeting program.

(ii) Without the patches in the Applewriter programs a syntax error in a DOS command enabled by control-D will land you in Basic. TEDITOR and PRINTER substitute their own error routine by replacing the error handler address normally stored at $9D5A/B. This has been relocated to $DD5A/B: hence the patches.

(iii) If (like me) you have a lower case adapter and have patched TEDITOR, then adjust HELLO accordingly. (Thus my version of TEDITOR is $1070 bytes long, so that line 36 of HELLO/SOURCE reads: LDA #$72.)

(iv) HELLO uses the Applesoft string printing routine STROUT and so is suitable as it stands for the Apple II Plus: for the Apple II it needs to be partly rewritten.

When DOS 3.3 is booted it bruns DOS MOVER as its greeting program. DOS MOVER relocates DOS, which then bruns HELLO as its greeting program. (See reference [2].)

HELLO bloads TEDITOR at $9D03,
bloads PRINTER at $AE03,
sets up &-vector to call $9C1B,
duplicates TEDITOR at $803 and calls it.

& calls HELLO at $9C1B to determine which program is resident at $803, duplicates the other one at that location and calls it.

If you wish to go from TEDITOR to PRINTER without the delay of loading it, exit the editor and type '&', (return): the printer program will come up almost instantaneously. The same procedure will take you from PRINTER to the editor.

MEMORY MAP

3F5 .. &-vector

803-1842 .. TEDITOR/PRINTER

Page DD upwards .. relocated DOS

Page BF .. subroutines to manage relocated DOS

AE03-BE42 .. PRINTER

9D03-AD42 .. TEDITOR

$803-1842 .. TEDITOR/PRINTER

3F5-3F7 .. & vector

The listing for the program HELLO, which does all the work, follows below.

REFERENCES

[1] Cornelis Bongers' program DOS MOVER is in CALL A.P.P.L.E., July/August 1981.

[2] Cornelis Bongers, 'Notes about DOS MOVER', CALL A.P.P.L.E., November/December 1981.

```
SOURCE FILE: HELLO/SOURCE
003C:                1 A1L      EQU  $3C
003E:                2 A2L      EQU  $3E
0042:                3 A4L      EQU  $42
03F5:                4 ANDVECTOR EQU $3F5
DB3A:                5 STROUT   EQU  $DB3A      ;APPLESOFT STRING PRINTING ROUTINE
FE2C:                6 MOVE     EQU  $FE2C
0000:                7 *
----- NEXT OBJECT FILE NAME IS HELLO
9C00:                8          ORG  $9C00
9C00:                9 *
9C00:A9 59          10          LDA  #>TEDNAME ;LOW BYTE
9C02:A0 9C          11          LDY  #<TEDNAME ;HIGH BYTE
9C04:20 3A DB       12          JSR  STROUT
9C07:A9 72          13          LDA  #>PRNNAME ;LOW
9C09:A0 9C          14          LDY  #<PRNNAME ;HIGH
9C0B:20 3A DB       15          JSR  STROUT
9C0E:A9 1B          16          LDA  #>CHANGE  ;LOW
9C10:A2 9C          17          LDX  #<CHANGE  ;HIGH
9C12:8D F6 03       18          STA  ANDVECTOR+1
9C15:8E F7 03       19          STX  ANDVECTOR+2
9C18:4C 37 9C       20          JMP  MOVETED
9C1B:AD 04 08       21 CHANGE   LDA  $804
9C1E:C9 F8          22          CMP  #$F8      ;IS TED AT $803?
9C20:D0 15          23          BNE  MOVETED   ;NO: BRANCH TO MOVE TED DOWN TO $803
9C22:20 4C 9C       24          JSR  DEST      ;YES: MOVE PRINTER DOWN TO $803
9C25:A9 AE          25          LDA  #$AE
9C27:85 3D          26          STA  A1L+1
9C29:A9 42          27          LDA  #$42
9C2B:85 3E          28          STA  A2L
9C2D:A9 BE          29          LDA  #$BE
9C2F:85 3F          30          STA  A2L+1
9C31:20 2C FE       31          JSR  MOVE
9C34:4C 03 08       32          JMP  $803      ;CALL PRINTER
9C37:20 4C 9C       33 MOVETED  JSR  DEST
9C3A:A9 9D          34          LDA  #$9D
9C3C:85 3D          35          STA  A1L+1
9C3E:A9 42          36          LDA  #$42      - 64
9C40:85 3E          37          STA  A2L
9C42:A9 AD          38          LDA  #$AD
9C44:85 3F          39          STA  A2L+1
9C46:20 2C FE       40          JSR  MOVE
9C49:4C 03 08       41          JMP  $803      ;CALL TED
9C4C:A0 00          42 DEST     LDY  #0
9C4E:A9 03          43          LDA  #3
9C50:85 42          44          STA  A4L
9C52:85 3C          45          STA  A1L
9C54:A9 08          46          LDA  #8
9C56:85 43          47          STA  A4L+1
9C58:60             48          RTS
9C59:0D 04          49 TEDNAME  DFB  $D,$4
9C5B:C2 CC CF       50          ASC  'BLOAD    TEDITOR,A$9D03
9C5E:C1 C4 A0
9C61:D4 C5 C4
9C64:C9 D4 CF
9C67:D2 AC C1
9C6A:A4 B9 C4
9C6D:B0 B3 A0
9C70:0D 00          51          DFB  $D,$0
9C72:0D 04          52 PRNNAME  DFB  $D,$4
9C74:C2 CC CF       53          ASC  'BLOAD    PRINTER,A$AE03
9C77:C1 C4 A0
9C7A:D0 D2 C9
9C7D:CE D4 C5
9C80:D2 AC C1
```

```
9C83:A4 C1 C5
9C86:B0 B3 A0
9C89:0D 00        54          DFB   $D,$0
9C8B:             55 ********************************

*** SUCCESSFUL ASSEMBLY: NO ERRORS

  3C A1L              3E A2L              42 A4L              03F5 ANDVECTOR
9C1B CHANGE         9C4C DEST           9C37 MOVETED        FE2C MOVE
9C72 PRNNAME        DB3A STROUT         9C59 TEDNAME

  3C A1L              3E A2L              42 A4L              03F5 ANDVECTOR
9C1B CHANGE         9C37 MOVETED        9C4C DEST           9C59 TEDNAME
9C72 PRNNAME        DB3A STROUT         FE2C MOVE
```

Further comments and additions added by John Rodger and John Sharp

The program submitted by Michael Clark illustrates one of the many advantages of having DOS in the Language (or RAM) card. Some further modifications and additions could further speed up your work with Applewriter. Michael's program uses the Ampersand vector to switch the alternative program into its normal operating position but you must first exit to Basic with the (Q)uit option. Some patches are given to allow the swap to occur automatically on choosing the (P)rint option in Teditor or the (R)eturn to Editor option in Printer. This also allows the ampersand vector to be altered to jump to $803 to allow reentry to the CURRENT Applewriter module after (Q)uit.

A further enhancement prevents the Printer module Bloading Print Constants every time you choose Print New Document and saving them again if you make a change. This is accomplished by adding a few additional bytes onto the beginning and end of the binary hello program to cause Print Constants to be loaded into Zero Page at the same time as loading in Teditor and Printer. Printer must also be changed slightly to inhibit the automatic Bload and Bsave. However you should always check and set the values of the print constants after initial entry from Basic as Applesoft uses the same locations ($50-$59) and is guaranteed to alter several of them. Once you are set up and switching between modules or using DOS commands within Applewriter then no further changes are required.
One further note- Do NOT try to enter the binary Hello program without DOS in the Language Card as you will overwrite your standard DOS.

Here are the additional patches

Teditor:-
128C: 4C 1B 9C (was 20 28 13)
(Moves Printer in on selecting "P")

Printer:-
100D: 4C 1B 9C
(Moves Teditor in on selecting "R")

B54: EA EA EA
AFB: EA EA EA
(Prevents additional Bloads and Bsaves of Print Constants)

Hello:-
(Here is a complete HEX dump of modified version suitable for both upper and most lower case versions of Teditor)

```
9BF9- A9 8B A0 9C 20 3A DB
9C00- A9 59 A0 9C 20 3A DB A9
9C08- 72 A0 9C 20 3A DB A9 03
9C10- A2 08 8D F6 03 8E F7 03
9C18- 4C 37 9C AD 04 08 C9 F8
9C20- D0 15 20 4C 9C A9 AE 85
9C28- 3D A9 42 85 3E A9 BE 85
9C30- 3F 20 2C FE 4C 03 08 20
9C38- 4C 9C A9 9D 85 3D A9 86
9C40- 85 3E A9 AD 85 3F 20 2C
9C48- FE 4C 03 08 A0 00 A9 03
9C50- 85 42 85 3C A9 08 85 43
9C58- 60 0D 04 C2 CC CF C1 C4
9C60- A0 D4 C5 C4 C9 D4 CF D2
9C68- AC C1 A4 B9 C4 B0 B3 A0
9C70- 0D 00 0D 04 C2 CC CF C1
9C78- C4 A0 D0 D2 C9 CE D4 C5
9C80- D2 AC C1 A4 C1 C5 B0 B3
9C88- A0 0D 00 0D 04 C2 CC CF
9C90- C1 C4 A0 D0 D2 C9 CE D4
9C98- A0 C3 CF CE D3 D4 C1 CE
9CA0- D4 D3 A0 0D 00
```

BSAVE HELLO, A$BF9,L$AC

Final note:- The articles on DOS MOVER ,mentioned in the references, can be borrowed from the Basug Literature Library by sending a large stamped addressed envelope to our PO box. We considered putting the disk in the library. However, we cannot sell APPLEWRITER for a nominal fee even if modified. The DOSMOVER ROUTINE will be on a library disk soon. If you send for the articles after we have put it in the library, we will let you know.

APPLE MACHINE LANGUAGE CORRESPONDENCE

London N.W.3.

Dear David,

I was delighted to see that my review of Apple Machine Language prompted two of your readers to add their own comments (Readers' Letters - December 1981).

Firstly, may I correct David Durling, who suggests that I condemned the book from high up in a tower of professionalism. Although my business interests have involved me with computers for nearly a decade, I have been programming in assembly language for only about two years. My "training" consisted of a once-a-week evening class at the North London Hobby Computer Club with the excellent Dr. Mike Brinson, together with the reading of all of the books and articles that I could find. It is for that reason that I felt justified in looking back two years and imagining myself reading the Inman book as my first instruction manual.

I believe that one of the tasks of a reviewer is to establish whether the claims made for the book by its authors and publisher are justified. In fact, I agreed with David Durling and Graham Dane and said in my article that "the book succeeds in its attempt to be a simple introduction ....". However, much more was, in my opinion, misleadingly claimed for it and such presumptiousness deserved exposure and condemnation.

The writing of well-designed and well-constructed computer programs is a skilful art and, as we all know, there are more than enough slapdash, sloppily-written programs around. We should all, by critism, complaint and refusal to buy, do our best to discourage the proliferation of any more. Why should the same principles not apply to "Apple Machine Language" ?

David Durling also asked me for recommendations for further reading. Writing in assembly language depends, to a much greater extent than in Basic, on an understanding of the internal workings of the computer. One of the best books that I have read on that subject is "Microprocessors for Hobbyists" by Ray Coles (Newnes Technical Books £3.25), which is now stocked by those branches of W.H. Smith selling Uncle Clive's ZX81. It does not relate specifically to the 6502 but the operating principles discussed are similar enough.

"6502 Assembly Language Programming" by Lance Leventhal (Osborne / McGraw-Hill) is a comprehensive reference manual but of less use as a tutorial.

I found Robert Findley's "6502 Software Gourmet Guide & Cookbook" (Scelbi) very readable and it contains a number of well written and carefully explained routines for mathematics, conversions, searching and sorting. As a second choice, there is Rodnay Zaks' "Programming the 6502" (Sybex) which I found irritating in that there are no "recommended solutions" to the exercises set in the text, many of which will be needed by aspiring 6502 programmers.

I have written a separate review of Randy Hyde's new book "Using 6502 Assembly Language". (EDITOR - Due to space constraints this has had to be held over to the next issue)

For my own part, I find that one of the best ways to learn is the "hands-on brain-bashing session". Although harder than copying solutions from a book, knowledge is gained by understanding and so sinks in deeper. Try converting some Basic programs into machine-code !

Kind regards,

Ian Trackman

5¼" DISCS BASF

PERFECT FOR APPLE II AND APPLE III

1 box (10 discs) **£16** per box
10+ boxes **£15** per box
100+ boxes **£14** per box

Post to: **SYSTEMICS LTD., 21-23 The Bridge, Harrow, Middlesex HA3 5AG.**

Order form Please send me:

Qnty.	Phone orders: 01-863 0079	Price	VAT	TOTAL
	Boxes of 10 5¼" DISCS			
	add £1.00 Delivery			£1.00

VALUE OF ORDER £

NAME

ADDRESS

Signed Phone

My Access/Barclaycard number is:

SEIKOSHA GP-100A Graphic Printer

80 cols, adjustable 10" width
Graphics, double-width chars,
Parallel Interface for Apple][& ITT

The GP-100A Uni-hammer graphic printer is a masterpiece in modern technology. The design of the future is here today.

ORDER NOW! FREE DELIVERY UK

full spec from :
MASS MICROS
WELLSON HOUSE, BROWNFIELDS
WELWYN GARDEN CITY AL7 1AR
TEL: 07073 - 31436 TELEX 298641

ONLY £215! + vat

RING FOR AVAILABILITY
WELWYN GARDEN 31436

MASS MICROS

GP *One Step Advanced!*
The GP series of Seikosha printers opens up a whole new world of low cost graphics printer

NOW YOU HAVE AN APPLE MICROCOMPUTER YOU'LL NEED

NEW

OMNIS®

All you've ever wished for in an information management system.

OMNIS sets new standards in database programs and levels of performance that you never believed were possible on a microcomputer

- **OMNIS** is written in UCSD Pascal+, this means a better structured, faster running set of programs than could ever be possible using Basic — We believe that UCSD Pascal+ is the best microcomputer language available — OMNIS proves it —
- **OMNIS** is structured around powerful file handling modules. These modules give you the flexibility to store and retrieve your information in the way that you want. Full multi-key indexed access is available to all your database files, you say what you want — OMNIS does the rest.
- **OMNIS** provides you with a versatile report generating module that enables you to define your own reports, lists, mailing labels etc.
- **OMNIS** has unparalleled search facilities to allow you to be selective. Those hours of fruitless searching through rows of card indexes becomes a thing of the past.
- **OMNIS** lets you design your own screen layouts for data entry and inspection — you may have up to 10 screens per file.

OMNIS has an application waiting for it in every business, school and laboratory and workshop. Wherever information needs to be stored and retrieved. OMNIS is available for both APPLE II and APPLE ///. We can also supply OMNIS for use on APPLE microcomputer networks (yes, with true multi-user record locking). Trade enquiries welcome

All registered users of OMNIS will be sent **FREE BACKUP** disc and you will be kept informed of all updates and upgrades. Free help will be given to all registered users via an OMNIS hotline.

OMNIS — All you ever wanted

APPLE II* version - £174.00 (incl VAT & pp)
APPLE ///* version - £225.75 (incl VAT & pp)

BLYTH COMPUTERS LIMITED
Wenhaston, Halesworth, Suffolk
IP19 9DH

☎ 050 270 565
24 hour phone service

*trademark of APPLE Computer Inc.
+trademark of the Regents of the University of California, San Diego

We welcome Access BARCLAYCARD VISA

® Registered Trade Mark

To: Blyth Computers Ltd., Wenhaston, Halesworth, Suffolk IP19 9DH
PLEASE SEND ME ☐ APPLE II VERSION ☐ APPLE /// VERSION
Cheque enclosed or please charge my Access*/Barclaycard no.:
*Delete as applicable
Name (Block Capitals)
Address
Tel. No

H1

THE LAST NOUGHT POINT EIGHT

by Cliff Wooton

IF you can dash off a BASIC program of professional quality in ten minutes flat, straight from your head,

OR have a beautiful set of routines already organised for inserting into your new programs so that they almost write themselves

THEN collect £200, pass jail, and

GOTO the final paragraph of this article
ELSE READ on

Recently, or rather some months ago, I set out to write a VAT program. This shouldn't take too long thinks I. After all, it's only a simple equation (VAT = NETT COST x VAT RATE/100)

There are a couple of complications to this basic equation in that there are currently 2 VAT rates (zero and standard) but a program needs to allow for up to 4 rates (zero, lower, standard and higher). Still not a problem. It also needs to allow for different inputs as sometimes a bill shows only the gross cost and you have to calculate the VAT from that rather than from the net cost. OK! Still no problem.

So a little program of about 20 lines will allow input of data, store it in an array, calculate the missing bits and print it out.

This program currently occupies 105 sectors of disk. It is still not finished and I have run into problems of delays for garbage collection.

So where has all that program come from?
Well, first of all, I am not the world's greatest programmer so no doubt a lot of it could be condensed by more efficient coding. Secondly, I write BASIC code as I go, direct from the keyboard without bothering about flow charts, etc., (doesn't everybody?), so the program flow could undoubtedly be improved. Thirdly, I am not a typist so the number of incorrect entries when inputting data requires fairly comprehensive input verification. It is not unknown for me to press a comma instead of a decimal point, or a letter where a number is required, so any program with a fair amount of data inputting has to be as bombproof as possible, otherwise it is quicker to do the whole thing with a pocket calculator and a pencil.

Finally, I am lazy and disorganised. I want to be able to input my invoices as I find them and not have to worry about getting them all into date order, checking the right date for the VAT period,, etc. So there has to be a means for deleting, inserting, altering data already input and for sorting into some sort (pun) of sequence.

Again, I do it all wrong by merrily writing editing routines as I go along instead of having (as YOU have, of course) a set of standard subroutines all ready to slip into my program.

So that program is currently in limbo awaiting time to get to grips with it and to turn it into something useful. By which time I will probably have forgotten the "clever" tricks I have used to overcome a specific difficulty.

Last week a guy at my local computer club, who is a wizard with machine code and claims that BASIC is too complicated for him to understand, sought my advice (which shows he is not as clever as I thought) on his attempts to write a simple bank balance program in BASIC. He reckons he could do it in machine code in half a day (and I believe him) but he feels that he should get to grips with BASIC as everybody tells him how easy it is.

Now! What could be simpler than a bank balance program? All it is is "NEW BALANCE = OLD BALANCE +- TRANSACTION"

He is, however, a little fanatical about the elegance of the inputting and outputting formats. He wants the resultant program to be foolproof and professional. Don't we all. Determined to show him how easy BASIC is and how misguided he is to prefer machine code for a problem which is tailor-made for BASIC, I casually and misguidedly volunteered to "dash off" a simple BASIC program which he could then study to see how simple it is.

The calculation bit is 20 lines, the inputting, verification, editing, replacing, deleting, formatting ....., is about 200 lines and still growing. I have spent probably 20 hours at the keyboard.

Of course, I should have copied chunks of my VAT program into this one, shouldn't I?

Unfortunately:-

a) I still have to get the VAT ones working properly

b) They have not been designed to be extracted as self-contained units

c) I didn't think about it before I started so I didn't "design" the bank balance program to allow this.

I now have two sets of non-working routines which do (or do NOT do) the same things.

Now I know that all of this is stupid and I am probably the only idiot who would do things this way (am I not?). For some time now, a glimmer has been developing in the depths of my subconscious that I seem to be writing the same routines in every program associated with business applications, or almost every program involving input of money. The bit that is specific to an application is usually quite small. Input and output page formatting is also quite simple once the data has been edited and given the right number of decimal places.

What I need, I think, is a standard program containing all of the possible input, sorting, recalling routines, to which I can simply add my little bits of calculation, page formatting, etc. I will have to rely on you believing that I came to this conclusion through my own brilliant deductive thought processes because, at the SGM on 22nd November, I bought a copy of Call-A.P.P.L.E. In Depth No 1 entitled "All About Applesoft". This contains a "program" consisting entirely of such routines. The idea is that you keep this program on a disk, just as it is. When you are inspired to write another masterpiece you simply load this lot into memory, delete any routines you are SURE you won't need and start from there. It also contains routines which avoid filling the string space and causing the dreaded garbage collection hang-up.

While it won't write your program for you, it should cover about 80 per cent of it - hence the uninspired title of this article.

The routines are claimed to be just about the best currently available for the various functions, certainly they are far more wieldy than those which I have cobbled together as I go along, and if you know a better one you can simply substitute yours in the master copy.

I believe that these routines (and the many others in the book) will be available from our library as soon as the disc is received from the States. There is, however, a vast amount of further info on Applesoft pointers, etc., which I presume will not be on the disc and which is necessary to obtain maximum benefit from the routines. The articles in the book are not simply collections of articles previously published in Call-A.P.P.L.E., but are specially prepared.

If you want a first class Xmas present ('cos it costs £8.95 and worth it for the above routine alone) you had better make the appropriate noises without delay as I am sure that this will be a sellout.

EPSON PRINTER PAGE

by John Sharp

The article I wrote on getting enhanced mode with the EPSON printer and APPLEWRITER has evoked a great deal of response. A number of members have found it useful as you may see from the letters column, and it has been used with other word processors. However, there have been some problems with some machines, but more of that in a moment. In the meantime there is obviuous need for a Special Interest Group on the EPSON. It is a superb printer, but the documentation leaves something to be desired. Fear not help is at hand. There will be a book published on how to use your EPSON soon, and we will keep you informed. The last communication I had with the publisher, suggested it had not yet gone to the publisher, although intially it was due out in January.

I do not have the time to write this column, and work on the machine, so is there a volunteer to coordinate the various letters and write this page. I can help with feeding information to you.

Let me begin with a little information. The EPSON you have may be running with the card supplied by EPSON, or you may have a standard Parallel card, or you may have the GRAPPLER card from Orange Micro. This allows you to do more with the graphics, such as dumping a picture sideways. Perhaps, one of you (Noel Padley ?) would care to review the facilities of this card. This sideways dumping is helpful in order to get the full enlarged picture on the MX-80. It would also be a help to others if you are not using the standard EPSON card, to let others know the benefits/disadvantages of using these other cards.

If you have the standard EPSON card, then look on the card to see if the PROM says A,B,C,D or whatever. If you have version C then you may have trouble. There are certain standards for software, such as having locations on the cards with numbers in them which can be recognised by the software so that it can find out whether it is a disk controller, parallel printer card or whatever. The version C of this Prom on the card was not standard, so that in particular PASCAL would not recognise it as a printer card. So if you are writing in with problems, not only do you need to say what the card is you are using, but which revision PROM is on it if it is the EPSON one.

Secondly, those who have the TYPE 2, i.e. the graphics version, have the same problem. There are new ROMS for the graphics and these have changed. There was a bug with the MX-80 and the new ROMs in that a dump of the graphics pages in the enlarged mode, caused an exit to monitor when it came to the end of the first line. It also gave trouble the first time you dumped a picture. There was a space between each scan on normal dumps. so, again if you write in say what type of graphics roms you have got.

Having said all this, it is hardly supprising that a routine that works on my machine, will not necessarily work on others.

Ray Harris, John Maltby and others found they obtained line feeds with the APPLEWRITER routine. The reason for this is they have the B version of the ROM on the card. I have one of the B ROMS but have not yet had time to modify my program to work with it. Another problem with the B ROM is that some of the programs on the PRINTER UTILITIES DISK 35 in the library, do not work properly with the B ROM. If you want upgrading and you bought an EPSON from BASUG, we will upgrade for a nominal #3, on an exchange basis. In order that you are not without your printer, send the money and a promise to return the B ROM. If you went elsewhere, go back to your dealer, and in the last resort come back to us. What I am not sure of was, did they have the data bit set in the Print Constants. Odd things can happen if it is not set such as spaces in between letters of words.

Shaun Hope pointed out some errors in my article. He said that line 10 should be:
10 PRINT CHR$(4);"PR#1";CHR$(13);: REM TURN PRINTER ON

There are two alterations here. The first to allow access through DOS is the CTRL-D (i.e. CHR$(4)). I deliberately left this out, and have done so in most of the programs I have written, because I had some trouble when I left it in (although it should be there if you go by the book), and have not had any since leaving it out. Unfortunately, I cannot remember what the trouble was. Again this may be due to the B vs D ROM differences, and indeed may even have been when I had a B ROM.

The second alteration is a mistake on my part. As Shaun says :-
"The important thing here is the Return Character (chr$(13)) after the disk command to switch on. Although this is mentioned in the Epson Manual, it is not exactly 'emphasised'. Without it you will not get very far. It is not needed on the switch off command."

In the case of the new graphics ROMs there is a problem with underline. As Colin Tripp pointed out, the example in the manual will give

under-lined printing.....

instead of correctly underlining as the example showed. This is a fault of the new ROMS, and yet another way to know which ones you have got.

The last problem for the moment with the EPSON card concerns VISICALC. Alan Jackson has pointed out that when you print out a file and then try to recalculate you cannot do so; you have to do an emergency save and then reboot and reload your data file. The reason for this is that the program which resides in the ROM on the EPSON card, uses some of the area in memory to store variables. This is why you POKE 1657 etc. The only way to overcome this is to use another card such as the GRAPPLER.
Please write in and say whether you got the routines to work, as I have had no trouble. If you do write in let us know what the system you have got in terms of cards, ROMs etc., it is just not as simple as saying it is an EPSON.

So to conclude...HELP. Will someone volunteer to coordinate this page in future.

AUTOCRAT COMPUTERS

SPECIAL OFFERS

QUALITY 5.25" DISKS
10 for £15.49+VAT

> EPSON MX80 FT <
>> £349 + VAT <<

Post & Packing Free

SPECIAL ARRANGEMENTS ON APPLE HARDWARE FOR BASUG MEMBERS

AUTOCRAT COMPUTERS,
264 Preston Road,
Harrow, Middx., HA3 0PY.

Tel: 01 908 3636

SOFTLIB EVALUATED

BASUG DISK 32: MATHEMATICS AND STATISTICS

Reviewed by R.D. Purves

This disk contains the impressive total of 62 programs, covering many of the common statistical calculations and text-book numerical methods. If the quality of the programs lived up to their quantity and variety, BASUG members would have access to a handy little library of mathematical software.

Unfortunately, most of the programs are just awful. Some do not work at all, or produce such strange results that their errors are immediately made apparent. Much more seriously, many programs produce plausible answers for a range of inputs, but give highly inaccurate results outside it. The statistical routines are so bad that ANYONE USING THEM WILL BE LED TO THE WRONG STATISTICAL CONCLUSIONS.

There is no documentation of the numerical methods used. In itself this would be enough to render the programs valueless, even if they used good algorithms, were efficiently coded, and contained few mistakes. In fact, nearly every program carries unmistakeable signs that its creator was deeply ignorant of the principles of numerical computation. Comments on a selection of the routines follow:

BINOMIAL DISTRIBUTION
prints a jocular misleading message when, owing to its feeble algorithm, it is unable to proceed with the calculation without risking numeric overflow.

CHI SQUARE DISTRIBUTION
fails completely for an even number of degrees of freedom, and gives absurd answers for an odd number.

F DISTRIBUTION
uses a grossly inaccurate approximation, as even a cursory check against published tables shows.

GAUSSIAN QUADRATURE
uses the 20-point formula, coded in an unusually inefficient manner. Two of the precomputed abscissae have been entered wrongly, which accounts for the program's inability exactly to integrate the function y = x * x.

MEAN VARI STDR DEVIA
manages to hit on the least accurate method of calculating sample variance and standard deviation. Poorly written statistical routines tend to fall apart when given data with a small coefficient of variation. My standard test for pocket calculators is to ask for the sample variance of (2000, 2000.1, 2000.2), the exact answer being 0.01. This program gives 6.6E-3, a remarkably bad result, in view of the excellent round-off behaviour of Applesoft's floating point arithmetic.

NORMAL DISTRIBUTION
contains a STOP statement which interferes with the smooth flow of information. After deletion of the offending word, the program is revealed to be about averagely OK. There are more accurate approximations to the normal distribution, and the coding is untidy, but this is one of the few usable programs in the suite.

PERMUTATION COMBINATION I & II
are identical apart from window-dressing. Both give the wrong answer for the no. of combinations of 0 or 1 item out of N. Neither can calculate (e.g.) the no. of combinations of 28 items out of 31 (answer = 4495), because the primitive algorithm is at risk from numeric overflow. NUMBER COMBINATIONS corrects some of these faults, but fails for combinations of 0 items out of N (the right answer is 1).

POLAR EQUATION PLOT
may be a clever and useful routine. Since there are no instructions I have no idea what it is supposed to do. It runs "silent and deep", then prints a row of asterisks.

ROOTS OF POLYNOMIALS
gives a REDIM'D ARRAY error if used twice. A purist would criticise the method used for evaluating polynomials, which is both slower and less accurate than Horner's nested parenthetical scheme. However there is a worse fault: roots already found are not divided out, and so you will be lucky to find all of them. Usually the program keeps finding the same root over and over again.

SIMPSON'S RULE
has an eccentric twist to it. Instead of asking for the number N of subdivisions of the interval of integration, it requests an increment DX, and then attempts to calculate N = (B - A)/DX. A check is made that N is an integer. Usually it is impossible for the user to find a suitable value of DX. In the unlikely event of a successful choice of DX, the program will often give wrong answers because no check is made that N is positive and even. This is a rubbishy piece of writing.

T DISTRIBUTION

claims to compute the right-tail probability. Actually it attempts to approximate a 2-tailed test, with disgracefully poor accuracy.

This is but a sample of the "riches" on the disk. Some programs have escaped my wrath merely by their paralysing banality (for instance HISTOGRAM, which prints rows of asterisks). There may be some reasonably sensible programs here somewhere, but I doubt if it's worth the effort of locating them. Certainly I would not use ANY of these programs, nor would I recommend them except as horrible examples.

The decision is up to the Software Librarian, but I think this disk should be withdrawn; it reflects no credit on BASUG. What should you do if you have already bought it? DOS provides a special command to cope with your problem; it's called INIT.

(John Sharp comments:)

Thanks for the review of Disk 32. You have highlighted the problem of managing the software library:-- We are simply offered too much software. We on the committee cannot hope to closely examine it all in haste and put our stamp of approval on it . However we feel it is better to make it available to members on an "as is" basis. Until someone like you speaks up and puts us in our place.

There is a crying need for documentation, and it is essential that members know where the programs came from. In this case they apparently mainly come from 'Some Common Basic Programs' by Poole and Borchers (Osborne). It would seem that a review of the book is called for as well. We can try to get a copy if you would do it although that may prove difficult. The disk was contributed from America.

The very real problem of documentation is being tackled, and we are considering withdrawing this disk, although perhaps we ought to continue making it available if accompanied by a warning that its use may endanger your health.

NEW! XMON

EXTENDED MONITOR

Written by Colin Richardson

XMON extends the Apple II System Monitor to offer a number of new commands as follows :

-S - single step
-T - trace
-M - improved move command
-X - exchange memory addresses (relocate without move)
-<CTRL-R> - relocate machine language programs
-<CTRL-F> - comprehensive find command
-P - hex dump with ASCII screen representation
-$ - hexadecimal to decimal conversion
-£ - decimal to hexadecimal conversion
-+ - 16-bit addition
-- - 16-bit subtraction
-* - 16-bit multiplication with 32-bit result
-/ - 16-bit division with 16-bit quotient and 16-bit remainder

Powerful new editing features are provided, and text page two can now be used in exactly the same way as page one. The new editing facilities are as follows :

-ESC-X delete character
-ESC-Y insert character
-ESC-Z copy complete line
-ESC-1 display text page one
-ESC-2 display text page two

In addition, the directional keypad cursor movements ESC-I, ESC-J, ESC-K and ESC-M which are only present in Autostart ROM Apples are provided for those with the original monitor. A Stop-List routine is also provided.

XMON comes with comprehensive documentation, making it both effective and simple to use.

£22.00 INCL VAT

MICROSOURCE
Tel: Park Street (0727) 72917

1 Branch Road,
Park Street
St. Albans.

EDUCATION COLUMN

By Norah Arnold

THE EDUCATION S.I.G.

I am happy to report that the Education S.I.G. has taken its first faltering steps into existence. Many thanks to those who answered the call. You will be hearing from me shortly if you have not done so already.

It appears that some teachers using Apple systems in their schools may be feeling that they are getting very little support from L.E.A.'s who are pushing Research Machines 380Z. Teachers of physics in particular would welcome more contact with other people developing physics software for the Apple. There is a possibility that the Education S.I.G. could assist by building up a library of physics software if enough people were willing to donate programs to get things started.

So how about it, you physicists? If you are interested but have not written in yet, contact me through the BASUG Box Number.

Despite the pressure from L.E.A.'s to acquire 380Z's it is obvious that where people have been free to make a choice, for example in our universities, many of them have chosen Apple as a tool to further their research in many areas of learning. Already we have made contact with several people working at postgraduate level who are using Apple II's. If there is anyone else out there who would like to join in and exchange information on their particular research speciality, contact me through BASUG.

From some letters received it seems that people who are writing software for primary age children often have difficulty in identifying and contacting schools where Apples are in use. Conversely, those trying to collect software for young children often cannot locate a programmer with whom to collaborate. If you fall into either of these categories why not take the plunge and join the Education S.I.G? We may, after all, be able to put people in touch with each other, to the benefit of both.

MICROCOMPUTERS IN PRIMARY EDUCATION.

Roy Garland's article on this subject in Hard Core No. 5 was most interesting. It does seem hard, as he remarks, that BASUG members can afford to buy Apple systems for themselves when most primary schools could not hope to acquire one without a great deal of parental fund-raising, which is not always forthcoming and in any case may have to be spent on basic needs.

I would like to take issue with him on one point, however. Primary schools are staffed in the main by women as he states, but this need not be seen as an obstacle to placing microcomputers in primary school classrooms. There are women computer 'addicts' working in primary schools (some have joined the Education S.I.G.), and it is these people who ought to be allowed to introduce women teachers to the micro, giving them an opportunity to see other women handling the machines with both confidence and competence. Cultural factors may seem to place males at the keyboard where leisure-time interest in microcomputers is concerned (perhaps they have more leisure), but in the workplace far more females operate word-processors, accounting systems etc. than do males.

I run a Computer Club for primary age children and find equal support from parents of both sexes. With the co-operation of her employer, one mother recently organized a trip to her workplace for a group of children. There she demonstrated the capabilities of the micro which she operates.

I do hope that the assumption by males that women teachers will not be able to cope with the micro does not prove to be the biggest obstacle of all.

APPLE MACHINE LANGUAGE.

This is not about education in the institutional sense but Apple users are continually educating and instructing themselves in the capabilities of their machines, so I make no apologies for accepting the Editor's invitation (Hard Core No. 5) to comment upon Ian Trackman's review of the book entitled 'Apple Machine Language' by Don and Kurt Inman. Personally I thought he was rather hard in his criticism. It seems obvious from the whole tone and appearance of the book, and in particular the cartoon-type illustrations, that it was meant for complete beginners. As such I found it both entertaining and instructive. Yes, there are mistakes in the book; tracking them down was fun.

It was not difficult to enlarge the small specimen routines given in the book into longer, more useful programs. All of my enlarged versions worked first time so the book must have taught me something!

More importantly I was able to go back and read with greater understanding books such as Lance A. Leventhal's '6502 Assembly Language Programming' and Leo J. Scanlon's '6502 Software Design'. This latter book also contains many mistakes, so 'Apple Machine Language' is not unique in that respect.

I would recommend any newcomer to Apple assembly language to start with the Inman's book. Don't stop with it, though, carry on to greater things!

(Editor - the controversy over Ian's review rages on! See Ian's reply to earlier comments elsewhere in this issue)

HELP!!

EDUCATIONAL PROGRAMS REQUIRED FOR DISABLED CHILDREN

Here is an opportunity for you to use your programming skills to help the less fortunate.

A local school for disabled children has recently acquired an APPLE and would greatly appreciate some help in writing some programs to help the children learn.

The school has given me details of what they expect the programs to do and in particular the type of input that the user is required to perform. It should be born in mind that keyboard input may not be possible in many cases.

Here are the proposed programs:-

1) Drawing Program.

Input Paddle0 and Paddle1 only

When PDL(0) is pressed a line is drawn on the screen preceeded by an arrowhead.
When PDL(1) is pressed the direction of the arrow will change by 45 degrees.
Arrow should start at centre of picture when program begins.
Choice of speed (1-10) (Presumably if plotting is automatically repeated?).

2) Numbers Counting

Input a) via keyboard
b) via two paddle buttons
A random number and pattern of objects such as balls, cars, houses, ducks etc to appear on the screen (Choice of maximum of 3,5 or 10 objects).
A box of numbers displayed at bottom of screen which is scanned automatically after a short pause on each. Range of numbers to be the same as the maximum number of objects that could be displayed. To stop the scan press PDL(0). Confirm by pressing PDL(1). Confirmation if correct etc. Option of number of attempts (3,5 or 10) before score is given.
An additional option to display a large number alongside objects if required would be desireable.

3) Number Snap

Screen to be divided into 2. Pictures (houses,cars,animals etc) appear on left. Large numbers on the right.
Two levels of play a) Numbers 1-5, b) Numbers 1-10.
If large number corresponds to number of objects then touch pushbutton. If correct then sound and picture to reinforce and congratulate. If incorrect or button not pressed then different sound and picture remains for a short period before continueing. Score after 10 changes.

4) Memory

A large number appears in centre of screen and remains for a short while before disappearing.
Input a) via keyboard -press correct number
b) via PDL buttons with a scanning box as above
c) via PDL dial- as dial is turned box moves forwards or backwards. Choice selected by pushbutton.
Options- 1,2 or 3 numbers appear at same time. After first number is requested then second and third in the same manner. For each correct answer display a happy face plus sound. For each incorrect answer display sad face plus no sound. After 10 goes score appears (number completely correct out of 10) and program stops.

If you can spare some of your time to do any one of these programs your efforts would be greatly appreciated. Send your programs on any media to John Rodger at BASUG address. If any efforts are duplicated then we will will send a selection to the school.
Thankyou in anticipation
John Rodger

P.S. The publication of this letter has been unfortunately delayed for two issues of Hardcore and early replies would be much appreciated by the school.

PALSOFT HIRES ENTRY POINTS

by Chris Chapman

The Hires entry points for APPLESOFT are documented in at least 2 places - Apple Orchard No 1 and Nibble 2.3 page 87. The data from these 2 sources differs except for DRAW, XDRAW and SHLOAD and the values given for the decimal and hex are not consistent! The PALSOFT entry points have not been published to my knowledge until now.

An annotated disassembly of the core routines in PALSOFT is available in the BASUG library but for most people the data given below should suffice.
The following abbreviations have been used :-
MSB = most significant byte
LSB = least significant byte
ACC = accumulator
Y REG = Y register
X REG = X register
$E6 = location for Hires page ($20 = page 1 and $40 =page 2)

HGR2

Enter at 62420 = $F3D4 ($E6 is set to $20 by the routine).

HGR

Enter at 62430 = $F3DE ($E6 is set to $40 by the routine).

CLEAR

Enter at 62446 = $F3EE (set $E6 to page required).

BCKGRND

Enter at 62452 = $F3F4 (the HPLOT routine must have just been used with the HCOLOR required for background).

HPOSN

Enter at 62456 = $F452 (the registers should be set up as :-
Y REG = MSB of x coord.
X REG = LSB of x coord.
ACC = y coordinate.)

HPLOT

Enter at 62616 = $F498 (the registers should be set up as for HPOSN).

HLIN

Enter at 62828 = $F56C (the registers should be set up as :-
Y REG = y coordinate.
X REG = MSB of x coord.
ACC = LSB of x coord.).

HFIND

Enter at 62968 = $F5F8 (the Hires cursor position is given by :-
$E0 = LSB of x coord.
$E1 = MSB of x coord.
$E2 = y coordinate.).

DRAW

Enter at 63028 = $F634 (the hires cursor should be positioned at the start of the shape using HPOSN. The registers should be set up as :-
Y REG = MSB of shape address
X REG = LSB of shape address
ACC = rotation
$E7 = scale).

XDRAW

Enter at 63023 = $F62F (the registers should be set up as in DRAW).

HCOLOR

Enter at 63177 = $F6C9 (the colour is set by :-
X REG = 0-7 (legal color values)).

SETROT

Enter at 63245 = $F70D (the rotation is set by :-
X REG = rotation (the routine stores the value in $E9)).

SETSCAL

Enter at 63251 = $F713 (the scale is set by :-
X REG = scale (the routine stores the value in $E7)).

SHLOAD

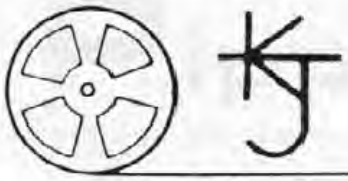
Enter at 63329 = $F761

DISPLAY1

Enter at 49236 = $C054 (displays HGR without clearing).

DISPLAY2

Enter at 49237 = $C055 (displays HGR2 without clearing).
Before using HPLOT , HLIN , DRAW or XDRAW remember to set HCOLOR to the colour required and GOOD LUCK.



apple computer
Sales and Service

kay-jay studios
135 COLES GREEN ROAD LONDON NW2 7HN
TEL: 01 450 6693

Finally you can realize the FULL graphic capabilities of your printer.

- Any area of the Hi-Res screen can be printed as large or small as you like.
- Zoom "Window" feature allows you to frame and see the specific area of the Hi-Res screen to be printed.
- Picture can be printed horizontally or vertically.
- Margins can be completely adjusted for custom placement of picture.
- Form feed and line feed can be controlled from the keyboard.
- Prints either Hi-Res screen 1 or 2.
- Prints charts, graphs or pictures.
- Menu driven . . . extremely easy to use.

ZOOM GRAFIX works with any of the following printers which have graphics capability: Anadex, Apple Silentype, Epson, IDS, Nec and Pro-Writer.

Only £29.95 inclusive of V.A.T., postage and packing.

THE NEXT ONE

The thinking person's program generator

Half price to B.A.S.U.G. Members at £50 + V.A.T. (15%).

THE NEXT ONE is a program which takes the drudgery out of program development. It eliminates the need for the the programmer to write all the tedious I/O subroutines between keyboard/disc, disc/core memory.
* The target (or generated) program is automatically structured and interactive with many diagnostic features built in.
* The target code is ready for use within a few minutes of creation by THE NEXT ONE.

THE NEXT ONE will write to a file on disc (named by the user) the program statements which comprise the program to be created.
After the file name has been chosen the data base array name and length are requested. This array is used for disc Input/Output.
Next; command, variable or array names are requested (up to 100 may be defined for the target program).
THE NEXT ONE goes on immediately to write code for the default commands and housekeeping subroutines for data handling.

Also available is a large range
of Structural and Engineering Software

Dealer Enquiries Welcome

Logical Computing,
26, Wide Lane,
Swaythling,
SOUTHAMPTON.
SO2 2HH.
U.K.

Verbatim®

MINIDISKS

Verbatim, the world's leading producer of minidisks, makes the widest range of minidisks available for word processing and data processing applications. Intent on maintaining this position of leadership, we constantly strive to make our minidisks even better.

Each Verbatim minidisk is equipped with a hub reinforcing ring to aid in registration, reduce errors and give better alignment repeatability.

Try new Verbatim disks for the performance of a lifetime.

SPECIAL OFFER TO B.A.S.U.G. MEMBERS ONLY

10 x MINIDISKETTES **£19·50** INCLUSIVE OF V.A.T. AND POST.

I ENCLOSE A CHEQUE MADE PAYABLE TO B.A.S.U.G. CHEQUE No ..... PLEASE FORWARD TO ME BY RETURN AT THE ENCLOSED ADDRESS.

NAME .....

ADDRESS .....

..... POST CODE

Pippin's Page ~~~~

Edited for younger readers by Vernon Quaintance

In December we started to learn about writing programs. Note that the usual English way of spelling programme is with two "m"'s and an "e" but when used in connection with computing it is always spelt the American way, as program. We also learnt about VARIABLES.

Do you remember the short program which was near the end of the last Pippin's Page? It was one to calculate the area of a circle of radius = 7. Let us see it again:

```
10 LET PI = 3.147
20 LET RADIUS = 7
30 LET AREA = PI * RADIUS
40 PRINT AREA
50 END
```

Type it in again. Don't forget to type NEW first to remove any old program that was in the Apple's memory. You might also like to clear the screen at this stage. Just type HOME <return> and see what happens. This command only works on the Apple. Other computers have commands like CLS to do the same job. If you want to, you can include a HOME command in your programs.

Now, supposing we want to be able to calculate the area of a circle with any radius? We must change the program so that RADIUS is not fixed at 7. We want the value to be put in from the keyboard, so we need a new command to do this. Have you entered the program above? If not, do so now and RUN it. Now type:

```
20 INPUT RADIUS
```

and LIST the program again. The first thing that you will notice is that the new line number 20 has taken the place of the old one. Whenever we wish to change a line, we can just type it all again and it will replace the old one. The other thing is only noticed when we RUN the new program. RUN it and see what happens.

Have you got a question mark on the screen? This means that the Apple is waiting for you to enter a number for the RADIUS. Try typing in a number now. As you press RETURN the Apple will give you the area of that sized circle. Now RUN the program again and enter a different value for the radius.

The problem with this program is that it does not tell you what is wanted when the question mark appears. Supposing you let your mother run your new program. She will not know what to do when the question mark appears, neither will she know what the number printed by the Apple is supposed to mean. Let us now change the program to tell her. Type in the changes shown below.

```
15 PRINT "THIS PROGRAM WILL CALCULATE THE AREA OF A CIRCLE FOR YOU."

19 PRINT : PRINT "PLEASE ENTER THE RADIUS ";

35 PRINT "THE AREA IS ";
```

Notice the trailing semi-colons at the end of lines 19 and 35. These stop the screen from starting a new line. The question mark thus appears on the same line as the question and the result follows the statement of what it is. Those of you who are observant will have noticed a colon (:) in line 19, between two PRINT commands. We can put more than one command on a line provided we tell the Apple when one ends and the next starts. This is done by separating them with a colon. Line 19 will therefore print a blank line before printing the request for the radius.

Try RUNning this program several times, with different values for the radius. Now LIST the program again. It should read:

```
10 LET PI = 3.147
15 PRINT "THIS PROGRAM WILL CALCULATE THE AREA OF A CIRCLE FOR YOU."
19 PRINT : PRINT "PLEASE ENTER THE RADIUS ";
20 INPUT RADIUS
30 LET AREA = PI * RADIUS
35 PRINT "THE AREA IS ";
40 PRINT AREA
50 END
```

We can now make two more changes to this program. Firstly, look at lines 19 and 20. These both relate to one input. Wouldn't it be nice if we could do both jobs with one command? Well, we can. Change line 20 to read:

```
20 PRINT : INPUT "PLEASE ENTER THE RADIUS "; RADIUS
```

Now delete line 19. How do you do that? Just type 19 <return>. Now LIST the program and then RUN it. It looks just the same on the screen doesn't it. Line 20 now contains a special type of INPUT statement (or instruction). This form is known as "INPUT with a prompt string". The words between quote marks (") in line 20 are known as a STRING because they consist of a number of characters strung together.

You have also just learnt how to delete a whole line of a program. Remember that when we type in a new line with the same number as an existing line, it replaces the existing line. If the replacing line has no commands after the line number then the old line will be deleted and nothing put in its place.

The second change also comes from combining two lines. Delete line 30 and re-type line 40 to read:

```
40 PRINT PI * RADIUS
```

Now re-RUN the program to prove that it still works in the same way.

Having to keep typing RUN to obtain the area of several circles of different radii is slow and boring. We can make the Apple go back to do it all again by means of another command. Type another new line as follows:

```
45 PRINT : PRINT : GOTO 20
```

Now RUN the program again. This time it keeps on going, asking you over and over for a radius and printing out the answers.

Enjoy your computing in 1982.

POSITION VACANT

An intelligent, well-spoken person is required to assist in the installation of Apple-based computer systems. The position will involve extensive travelling and offers excellent prospects to the right person.

Salary and Conditions are negotiable.

Applications in the first instance by telephone to 060 545 481.

RIP-OFF NEWS

No, this is not a thieves' kitchen - the opposite in fact. The idea is to set up an information exchange for the benefit of software writers - BASUG members - whose products are being copied and either sold or used in unauthorised ways. Whenever you come across blatant copyright violations of this kind you should drop us a line. We do not intend to publish this information but simply to pass it on to the authors concerned, if we can trace them.
We have to stick to the following ground rules:

Please don't tell us about individuals taking pirate copies for their own use. These individuals are very, very bad boys, and we heartily disapprove - but which of you, hand on heart...?
No, tell us about copying for commercial gain. Tell us who did it, when, where, how much, what the product was - all the relevant details. We would rather you didn't send anonymous tip-offs - better for you to give your name but indicate that you want to be kept out of it. On the other hand if you would be willing to stand as a witness in a possible prosecution, please indicate that too.
We would also be interested in hearing about customers who have bought in a product for one institution and then sent out duplicates to all sister institutions: say, a local education authority buys a package for one school and then puts it into a resources pool for all schools in the area without seeking permission. This may or may not be illegal, but where it happens the authors should know about it.
Tell us about user groups (including BASUG!) which have included proprietary software in their free software exchange catalogues.
My first inclination is to ask for us to confine this to members' products - a service for the little man - and let the big corporations look after themselves - but that is rather an arbitrary decision which needs thinking about. In general, please let us know what you think of this scheme. Is this an area BASUG should be in?
Let me repeat: BASUG is not going into the sleuthing business and we are not yet ready to set up a legal department to press charges on members' behalf. But we will actively promote the exchange of information.
Write to: Tony Williams, Rip-Off News: BASUG, P.O. Box 174, Watford WD2 6NF.

MICROSOFT TASC COMPILER

A review

by Peter Broadbent

Having bought a copy of Microsoft's TASC a few days before receiving the December 1981 issue of Hardcore, I was most interested in the feature on Applesoft Compilers, especially after a hurried glance at the author's reccomendation confirmed I had made the right choice. The article contains a wealth of information but without hands on experience I, at least, would not have appreciated all of it. Also, a balanced comparison of four compilers might miss specific points on each of them. In the circumstances I felt that the initial reactions of a user, together with some additional information on TASC might be of interest.

The production version of TASC is not protected and indeed it is prudent to work with copies as the disk runs continuously during compilation which takes about 100 seconds to process 1K of source program. Step 1 is to save the Basic program on the TASC disk and start compilation with the instruction RUN TASC. A series of prompts then ask for program name and addresses for components of the finished program. The process ends with the compiled (object) program saved on the TASC disk and with addresses of Runtime Library, Object Program and Allocation for Variables available on the screen. A table which lists object program addresses for each of the source program lines is also available. Finally the object program is BLOADed and the command EXEC ADR (ADR is included on the TASC disk) then results in a display of object program address and length to facilitate BSAVEing the permanent copy. A copy of Runtime must of course be similarly BSAVEed on each working disk. Accepting default addresses puts Runtime at 2051 followed by the object program at 6064 and this is followed in turn by the allocation for variables and arrays. Standard options locate the object program above HGR or above HGR2 but any address can be specified.

Some of the limitations of TASC are tabulated in Fig. 3 of the December article and others are that it will not successfully deal with a self modifying source program, e.g. Phone list, which is supplied with the Apple. It crashes if the total call on memory would exceed 48K and it might crash when proccessing a very long program due to internal tables becoming overloaded. However, the facility for passing variables from one program to another makes it practicable to run a very long program by breaking it down into sub programs and running through sequentially. I consider the author makes heavy going of the need to BLOAD RUNTIME and then BRUN OBJ.PROGRAM. A simple one line Basic program on the same disk sets everything up.

Reference to Fig. 2 of the original article implies that TASC is not particularly successful with the bubble sort program, Listing 4. A check closely confirmed the published performance as follows:-

TEST		LENGTH	SPEED C1	C2
C(1)		476	124	
C(M)	Runtime	4013		
	Object Program	546	40	94
	Variables & Arrays	534		

However, Listing 4 is not in optimum form for TASC which works faster if variables have to be declared individually but in a case such as Listing 4 a blanket declaration can be made in the form:-

```
1 REM! INTEGER*
```

In the case of individual declarations the variables are typed on the same line, separated by commas.

Adding the above line 1 to Listing 4 cuts the length of the object program to 511 and the allocation for variables to 216. The important point however is that execution time was reduced from 40 seconds to 16 seconds.

Test B, the check on the growth of object program resulting from the addition of an 8 byte line to the source program indicates a growth of 20 bytes (2.5X). I consider this gives a false impression of the size of object programs and, in fact, the data presented above for Listing 4 indicates a growth of 15%; Listings 2 & 5 have growth of a similar order. Test B reflects the high memory overhead required to deal with variables but when considering Listings 2, 3 and 5 it is necessary to take into account that REMs in a source program are ignored by the compiler. The need to accomodate the 4K Runtime Library has also to be kept in mind.

My own experience is that reasonably compact working source programs, stripped of REMs, have grown by about 50%. If there is a substantial proportion of text the increase might well be less. The worst case I have encountered is a 75% growth in a small program which was a relatively large array. Length of compiled programs has not proved to be a problem but I must admit that in certain cases this is because I concurrently found Cornelis Bonger's exellent program for relocating DOS 3.3 on the Language/RAM card** . The additional 10.2K of memory made available more than compensates for increased length of programs.

I have a suspicion of only one minor bug in that there have been two occasions when I have

attempted a compilation in the middle of a general session and there has been a problem with addressing the object program. Whilst I find it difficult to believe that data already in the Apple will corrupt TASC I now switch off immediately before a compilation and boot with the TASC disk. Otherwise TASC has behaved impeccably.

My main reason for buying TASC was that I have a suite of programs which call for entry of large amounts of data, each item being subject to a short processing routine before being stored and eventually sorted. The processing caused a barely perceptable delay which I nevertheless found irritating. There was no justification for spending time writing the programs in machine code - even if I was proficient. I now have the pleasure of seeing the prompt for the next entry thrown up before my finger leaves the return key and speed of sorting is vastly improved, by a factor of 12 in the only case where I have carried out a test.

** See Call A.P.P.L.E. July/August 1981.

HELP WANTED

MAGAZINE ... SOFTWARE ... MEETINGS ... LIBRARY ... EXHIBITIONS ... HELP & ADVICE ... SPECIAL OFFERS ... ETC ... ETC ...

Q. Where does all this stuff come from ?

A. From You - or some of you !

BASUG is a voluntary group, and all these services are provided BY, as well as FOR, the members themselves.

So if you can help in any way, please get in touch.

DON'T FORGET THE EXHIBITIONS
ALCC LONDON COMPUTER FAIR
15-17 APRIL

COMPUTER FAIR
EARL'S COURT
THE NEXT WEEKEND

SOFTALK

Softalk is THE APPLE magazine in the USA and up to now has not been available except to registered APPLE users in the USA only. By special arrangement, we can now offer subscriptions to you. As we hope to bring the price down, we are offering it on a six month subscription for £17.

It is the best value for money of the American APPLE magazines. It normally has 180-200 pages of articles covering a wide range of software and hardware for the APPLE. There are short listings, but mainly the accent is on specialised pages:- VISICALC, SOFTCARD, APPLE ///, PASCAL, BASIC, BEGINNERS' CORNER, ASSEMBLY LANGUAGE etc. plus reviews on all the month's new product releases, not just some but all!

FOR a trial issue, of the BUMPER CHRISTMAS EDITION, send £3 to MICROSOURCE at the address below.
FOR a six month subscription starting with the January issue, send £17.
If you want the Christmas edition, plus a subscription, send £19.

MICROSOURCE
1 Branch Road, Park Street, St. Albans.
Tel: Park Street (0727) 72917

CATALOG LABEL PRINTER

BY John Sharp

As you write more and more programs you build up a large stock of disks. The problem then arises as to where a particular program is. There are disk catalog prgrams like AUTO-INDEX and MASTER CATALOG which allow you to search and index your disks. There is still a need to have a label which goes with the disk. However, the long single column list normally obtained from a catalog is not too convenient for sticking on the envelope. The following program was written for another purpose and then modified. It was written to send a catalog to a printer on a VISTA GAS CHROMATOGRAPH. Those handful of people who know what I am talking about can contact me to obtain knowledge of how to send the information to the VISTA and print out on the VISTA printer. I needed to do it this way because this particular printer will only accept ASCII code under a particular handshaking routine.

What follows is thus a basic BASIC program which puts the Catalog of the disk into an array, and you can then manipulate the array to your own end. It also aims to help the BASIC programmers over the hurdle of the machine code transfer of the catalog into the array. For example the next step might be to arrange all the Integer programs to come first and then Applesoft and then Binary files etc., or you may want to sort on the filename, or have three columns instead of two.
It is written with constants for 3.3 DOS.

The program uses the RWTS routines as in the DOS manual (p 94 OOS 3.3). As this is explained fairly well and you do not need to understand it, I do not intend to describe it in detail. I would like to explain what happens when the catalog is in memory and is transferred to the array. If you want to see the explanation of the different parts of a catalog page, then read "BENEATH APPLE DOS".

Using the RWTS (which incidentally means Read Write Track Sector) routine it transfers the catalog into memory above the second graphics page. The actual sorting out takes a little time, so to ensure the program is carrying on OK and hasn't just hung, every time it goes through the loop to find a new page of tye catalog, then it will print a ">" on the screen. The final part of the profram separates the program into two halves to print side by side.

Since you may like to modify it yourself a more detailed explanation of the program is in order.

To begin at the beginning, with the first line. This sets up an array to hold the catalog entries. As written it puts the whole entry, i.e. whether it is locked, what type of file, the length and the file name, into a single string. This i s for convenience, as well as saving memory. This may be more useful if you are sorting, because you then do not have to worry about keeping all the parts separate, but more of that at the end. Any deleted catalog entries are ignored, although they could be found if necessary.
There is a machine code file "RWTS" which does the actual work. This is lifted straight out of the DOS Manual, and located to run at $2000, 8192 decimal. For the convenience of those who cannot load machine code, a POKE version is listed in the modification section.

The lines up to 230 then do the actual movement of the catalog into memory. Again if you are interested, it is written up in the DOS manual. The program as written, is for DOS 3.3 DRIVE 1, in SLOT 6. The DOS manual will show you what to alter for DOS 3.2 or for the other slot drives.

Now the real work is done. Lines 250 to 580 scan the memory sorting out, first whether it is locked (line 370) and then the type (lines 380-450). These types are in some cases rare and may not yet be in existance; they were taken from the book "BENEATH APPLE DOS". The codes 8 and 32 I have yet to see, but you are ready if they do turn up. The most uncommon one, "R" is a relocateable binary file, and you will find examples on the DOS 3.3 TOOLKIT disk; they are produced from the APPLE ASSEMBLER.

A word about lines 300 and 330 before I go on. These are inserted, because the program takes a little time if the catalog is long. They just show that something is happening. The ">" shows each track as it is searched, and the "." each file. It helps you to know that there is not a fault and it the machine just hanging.

The string NAME$(N) for file N is added together as follows.
Line 340 simply puts a blank space in to start.
Line 370adds a space if the file is unlocked and
Line 380 puts a "*" there if it is.
Lines 390-450 add the file type
Line 490 adds the length with spaces either side, after it has been adjusted to have the leading zeros added where necessary, in line 480.
Line 550 builds up each character of the file name.
Line 530 detects whether there are any more files left, and if not finishes.
Line 350 looks to see if a file has been deleted, and if it has ignores the title. The name and position is still there, and this information can be used to Undelete it. A possible future modification?

Now we are left with an array containing all the files in NAME$(N) where N is the number of files, to do with as you will. Lines 770 -1110 do the printing, in this case for an EPSON, with the file names and other information, two across. For those with another printer, line 620 sets the character width, and the CHR$(15) in line 630 sets upp condensed mode.

Modifications to the basic program.

1. With the references to lines 320-570 adding up the NAME$(N) it should be fairly easy for you to put the locked/unlocked, file type, file length and file name into a multidimensional array. This should be dimensioned in line 50 as INFO$(105,3) then
INFO$(N,0) could be loaded with " " or "*" depending on whether the file is locked or not.
INFO$(N,1) with the file type
INFO$(N,2) with the length and
INFO$(N,3) with the file name.

2.It would be easier to sort on file name alphabetically, or for file type, by using MID$(NAME$,X,1) rather than on separate arrays. I am not sure on the efficency, perhaps an article from a sorting expert would be in order.

3.The RWTS routine may be entered with the following pokes:-

```
130 FOR N = 8192 TO 8225 : READ M : POKEN,M : NEXT

132 DATA 169,32,160,10,32,181,183,96
134 DATA 0,0,1,96,1,0,17,0
136 DATA 27,32,0,33,0,0,1,0
138 DATA 0,0,96,1,0,1,239,216,0
```

Anyone any further ideas? Any more basic programs that can be built upon?

RWTS

```
2000- A9 20 A0 0A 20 B5 B7 60
2008- 00 00 01 60 01 00 11 00
2010- 1B 20 00 21 00 00 01 00
2018- 00 60 01 00 01 EF D8 00
*
```

LIST

```
100  DIM NAME$(105)
110  TEXT : HOME
120 D$ =  CHR$ (4)
130  PRINT D$;"BLOAD RWTS"
140  HTAB 15: PRINT "CATALABEL"
150  HTAB 8: PRINT "CATALOG LABEL
     GENERATOR"
160 ISECT = 8207:ICMD = 8214:IBUF
    P = 8211
170  REM  xxxx READ CATALOG INTO
     MEMORY
180  VTAB 10: CALL  - 868: PRINT
     "PUT DISK IN DRIVE AND PRESS
     RETURN": GET A$: IF A$ < >
     CHR$ (13) THEN 180
190 : TEXT : HOME : VTAB 10: HTAB
    13: PRINT "GETTING CATALOG"
200 CMD = 1:BUFP = 33:SECT = 15: POKE
    ICMD,CMD
210  POKE ISECT,SECT: POKE IBUFP,
     BUFP
220  CALL 8192
230 SECT = SECT - 1:BUFP = BUFP +
    1: IF SECT >  = 1 THEN 210
240  TEXT : HOME : VTAB 10: HTAB
    13: PRINT "SORTING CATALOG O
    UT."
250  VTAB 20
260  REM  xxxx FILE INPUT ROUTINE
     FROM MEMORY INTO ARRAY
270 B = 8459
280 N = 0
290  FOR R = 0 TO 14
300  PRINT ">";
310 A = B + 256 * R
320  FOR S = 0 TO 6
330  PRINT ",";
340 N = N + 1:NAME$(N) = " "
350  IF  PEEK (A + S * 35) = 255 THEN
     N = N - 1: GOTO 570: REM  xx
     x IF FILE HAS BEEN DELETED T
     HEN IGNORE
360 TYPE =  PEEK (A + 2 + S * 35)

370  IF TYPE < 128 THEN NAME$(N) =
     NAME$(N) + " ": GOTO 390
380 TYPE = TYPE - 128:NAME$(N) =
    NAME$(N) + "*"
390  IF TYPE = 0 THEN NAME$(N) =
     NAME$(N) + " T"
400  IF TYPE = 1 THEN NAME$(N) =
     NAME$(N) + " I"
410  IF TYPE = 2 THEN NAME$(N) =
     NAME$(N) + " A"
420  IF TYPE = 4 THEN NAME$(N) =
     NAME$(N) + " B"
430  IF TYPE = 8 THEN NAME$(N) =
     NAME$(N) + " S"
440  IF TYPE = 16 THEN NAME$(N) =
     NAME$(N) + " R"
450  IF TYPE = 32 THEN NAME$(N) =
     NAME$(N) + " B"
460 LGTH =  PEEK (A + 33 + S * 35
    )
470 LH$ =  STR$ (LGTH):L =  LEN (
    LH$)
480  FOR W = 0 TO 2 - L:LH$ = "0"
      + LH$: NEXT
490 NAME$(N) = NAME$(N) + " " + L
    H$ + " "
500  FOR M = 0 TO 30
510 E =  PEEK (A + 3 + S * 35 + M
    )
520 F =  PEEK (A + 4 + S * 35 + M
    )
530  IF E = 160 AND F = 160 THEN
     570
540  IF E = 0 AND F = 0 THEN NAME
     $(N) = " ": GOTO 600
550 NAME$(N) = NAME$(N) +  CHR$ (
    E)
560  NEXT M
570  NEXT S
580  NEXT R
590  PRINT
600  TEXT : HOME : VTAB 10: HTAB
     10: PRINT "SENDING CATALOG T
     O PRINTER"
610  PR# 1: PRINT  CHR$ (0)
620  POKE 1657,100
630  PRINT  CHR$ (15);"CATALOG"
640 N = N - 1
650 BD = N / 2 -  INT (N / 2 + .0
    02): IF BD > .1 THEN P1 = 1:
     GOTO 670: REM  xxx THERE AR
     E AN ODD NUMBER OF FILES
660 P1 = 0: REM  xxx THERE ARE AN
     EVEN NUMBER OF FILES
670  REM  xxxx NOW DO THE ACTUAL
     PRINTING
680  FOR Z = 1 TO N / 2
690  PRINT NAME$(Z); TAB( 40);NAM
     E$(Z +  INT (N / 2) + P1)
700  NEXT
710  FOR M = 1 TO 5: PRINT : NEXT

720  PR# 0: IN# 0: CALL 1002: REM
      xxx REHOOK KEYBOARD AND DOS

730  TEXT : HOME : VTAB 10: CALL
      - 868: INPUT "DO YOU HAVE A
     NY MORE TO CATALOG ";ANS$
740  IF ANS$ = "YES" OR ANS$ = "Y
     " THEN 160
750  END
```

Spider Software

Asteron – The best implementation of the Asteroids arcade game **£19.95**
Autobahn – Road race game with sound and hi-res graphics **£14.95**
Beer Run – Catch the Artesians before the Guzzlers and Bouncers catch you **£16.95**
Borg – Can you out run and outshoot the dragon's henchmen? **£16.95**
Copts & Robbers – An adventure in the tombs of ancient Egypt **£19.95**
Cranston Manor – A new hi-res adventure like Wizard and the Princess **£19.95**
Crossfire – Stop the aliens from taking over the city **£16.95**
Dark Forest – An adventurous game of strategy and conquest for up to 6 players **£16.95**
Epoch – 3-D hi-res game in space **£19.95**
Expediter II – THE APPLESOFT COMPILER AT A LOW, LOW PRICE **£56.95**
E-Z Draw 3.3 – The poor man's graphics tablet **£28.95**
Hadron – A 3-D battle in the midst of a meteor field **£19.95**
Jawbreaker – Eat up the dots but watch out for the Gobblers **£16.95**
Mission: Asteroid – Hi-res adventure in 21 colours. Save the World!! **£10.95**
Mystery House – Hi-res adventure using over 100 pictures **£12.95**
Multi Boot Upgrade – Upgrade 3.2 discs so that they can boot on a 3.3 Apple **£23.95**
Norad – Real-time hi-res simulation of a nuclear conflict **£19.95**
Outpost – Save your space outpost from attack by enemy fighters **£16.95**
Pascal Graphics Editor – Professional graphics editing package for Pascal **£49.95**
Pegasus II – The latest in pub games now available for the Apple **£16.95**
Phantoms Five – A fighter-bomber mission in real time hi-res graphics **£18.95**
Shooting Gallery – Real-time simulation of a fairground gallery **£15.95**
Space Warrior – Protect your shields from the dreaded drone RAM ships **£13.95**
Star Avenger – A fast paced game of guerilla warfare in space **£19.95**
Star Cruiser – Save yourself from the swooping aliens **£18.95**
SuperScribe II – 70 column word processor with true upper/lower case **£73.95**
The Dictionary – Checks spelling of any standard DOS text or binary file **£54.95**
The General Manager — User-friendly hierarchial database package **£54.95**
Threshold – Super fast arcade style game **£21.95**
Time Zone – 1,182 room hi-res colour adventure **£54.95**
Ulysses and the Golden Fleece – Another adventure like Wizard and the Princess **£19.95**
Wizard and the Princess – Hi-res adventure in 21 colours **£18.95**

A: Requires Applesoft in ROM.
M: Will run on any Apple II.

All programs will operate on an Apple II or II+ with 48K. All programs will operate under DOS 3.2 or 3.2 with the exception of E-Z Draw 3.3, Pascal Graphics Editor, The Dictionary, The General Manager and SuperScribe II which all require DOS 3.3.

We can also supply Visicalc, Wordstar, Z80 Softcard, 16K Ramcards, etc at very competitive prices.

WE ALSO OFFER A BESPOKE SOFTWARE SERVICE WHICH IS SECOND-TO-NONE

Prices include VAT at 15%. Add 50p p&p for orders under £30 totally.

Please write or telephone for your free copy of our up-to-date software list.

DEALER INQUIRIES INVITED. PERSONAL CALLERS BY APPOINTMENT ONLY PLEASE

Spider Software

98, AVONDALE ROAD, SOUTH CROYDON, SURREY.
Telephone: 01-680 0267 (24 hours a day — 7 days a week)

LABORATORY MICROCOMPUTER

For the scientist who:
uses a programmable microcomputer with his equipment, be it for lc, ms. nmr. exfas, particle size analysis, materials science,
or who
uses a stand alone microcomputer for scientific calculations

LABORATORY MICROCOMPUTER

is designed to become his own community forum, the voice of all who work in the physical, life and medical sciences and who are serious users of microcomputers

Appearing four times a year, LABORATORY MICROCOMPUTER will be edited by Dr Brian Millard, late of the Institute of Neurology, University of London, and currently involved in the design and utilisation of microcomputer systems for laboratory use. The yearly subscription has been deliberately kept low at £10 for personal copy subscriptions.

TYPICAL CONTENTS WILL INCLUDE
Laboratory aspects of hardware, software and peripherals
Problems and their solutions
Extended user trials
Products and services
Conferences and exhibitions - long range diary
Letters to the editor

The successful development of LABORATORY MICROCOMPUTER will establish an accurate channel of communications between the users and the providers of hardware and software, to the benefit of both groups.

.....

To: Science & Technology Letters, 12 Clarence Road, Kew, Surrey TW9 3NL.

I wish to subscibe to LABORATORY MICROCOMPUTER Vol.1 (1982) at £10 a year

Cheque/P.O. enclosed Institutions requiring an invoice please tick here

Name.....

Address.....

Signature..... Date.....

We shall be glad to send you a specimen copy if you tick here. ☐

reader's letters

Dear BASUG,

I am the owner of an Apple II plus with Centronics model 737.2 printer and Diplomat parallel interface. I would be very grateful if you could tell me how to get the Apple Writer program to print different character styles as shown in the Centronics owners manual. Also I would like to know if it is possible to print Apple graphics on the Centronics, and if so, how.

Thanking you in anticipation
Yours faithfully,

Peter Brown

(Ed. Your troubles are at an end. The magazine you are reading, enhanced bits and all, was produced on that very printer. You can switch to the various type styles
1) by running a small BASIC program setting up the characters and type styles before activating your Apple Writer program. This is limited in usefulness.
2) by doing like we do and talking to the 737 from within Apple Writer files using the Go-Between available from Apple. 3) By using Neil Lomas' APPLE SPEIL which is a BASIC word processor program which was specially written to cope with the CENTRONICS printer. If John Rogers hasn't put it in the library this time, he will no doubt get some angry letters from you as well as some comments from the rest of us on the committee.
Graphics? The 737 does not allow graphics and can't be converted-according to Centronics. The 739 is the version for graphics. No doubt with this prompting then someone will come back and say they have done it -so much the better, that is what BASUG is about.)

London SW 7

Dear Mr Sharp

I have had my Apple for about a year and I will be sixteen at the end of March. I am fluent in BASIC and I am learning Pascal and Machine Code. In between studies for Maths, Physics and Chemistry A-Levels I would very much like to do some programming for the club, for the experience rather than financial gain. If you know of any opportunities for me to do such development work I would appreciate it if you could let me know.

Yours sincerely

J.E.Shaw

(Ed. We have passed on a few ideas to Jonathon, but if any of you have any suggestions there are other members as well who have an Apple waiting for ideas.)

Chorley

Dear Sir,

... When I first joined in March, I got a disk (No. 8) which contained a program called "Death Race". Muffined to 3.3 it wouldn't work. Demuffined back to 3.2 it still wouldn't work. Cause? It uses the Programmers' aid. Solution? To those with Applesoft only, none. For those with either the language card or the Integer Board, get to Integer Basic, then BRUN and it works fine. Incidentally, this applies to quite a number of Programma Software stuff, characterised as being binary files and invariably 3.3. DOS.
Hope this is of some use to you. Another snippet of useful gen is to demuffin FID 3.3 to 3.2 for file handling. It's far better than the standard 3.2 file copiers and in my experience it works perfectly, whereas the FID 3.2 doesn't.

Yours faithfully,

Glyn Davies

(Well done Glyn, in sorting that out. There was some problem with the FID program on that disk (the file was corrupt on the original) but I think was sorted out. As you say the FID program is useable from 3.2 DOS and the version on the disk was only a copy done the way you did it.)

London

Dear Hard Core,

I have some items of game software with no documentation. Can anybody out there help? They are: i) Dogfight ii) AZFS1 - Flight Simulator, iii) Star Warrior. would any members care to give me some hints on a) Wizard and the Princess or b) Wizardry, or both? In a) I've got stuck after meeting the giant across the second bridge. In b) I have six characters, most of which are of level 13, but I can't seem to get the armor class of bishop below 5. Also, he's got only about 40 hit points which means that he's the first to go usually. Any suggestions would be appreciated, thanks

Miss S H Chuah

Highgate

Gentlemen,
... Sorry I can't offer an Apple Writer letter. I Use "Easywriter" (the cheap 40 column version) and an Epson MX-80 was considerably encouraged by John Sharp's article in October's Hard Core to find out that I was not alone in having difficulty with the manual. I don't, however, agree with his comment that you can't get enhanced mode from within word-processing programs. Watch this!

Reading some back issues of Hardcore, I saw a suggestion on some kind of 'Library' facility which might stop us all buying the same, expensive, magaznines. Did it develop? I appreciate that the copyright laws make it impossible to circulate copies, but what about an "abstract" or just the contents page. If I cold help I'd be happy to try.

Yours faithfully,

Charles Watts-Jones

(Ed.Something is happening on the abstract front, John Rodger will inform you via the library list. In the meantime since your word processed letter is being typed via a Centronics the readers will have to take our word that from "Reading some... "onwards, his letter was indeed enhanced. Now, to enhance my pleasure, please accompany unorthodox WP letters with standard format printouts which we can paste in.)

Bottesford

Dear Sir,

I note that you are intending to hold national meetings in London. As a Londoner myself ... it wasn't until I left the city that I became aware of the southerners' image of London as the centre of civilization and people should be expected to travel the length and breadth of the country in order to attend happenings there.... To be practical, what's wrong with Birmingham, I ask?

I like the idea of Hard Core being published on a commercial basis... I think it good and worthwhile reading. However, rather than putting up the membership fee for newcomers, what about dropping it for us olduns?
As an industrial interior/furniture designer ... I am becoming increasingly interested in the use of micros for computer aided design and feel that this is a subject not very much aired in the micro world at the moment. Incidentally, by CAD I means all computer aids for designers, not only drafting methods. I might be interested in organizing some speakers on this.

Kind regards,

David Durling
(Ed. Thanks for the offer. You know by now that BASUG has pulled out of the Windfall spectacular, but we have our own plans and hope to be able to call on just such speakers. We agree that NATIONAL meetings are our aim, and they should move around the country. The problem is who does the work and who pays the expenses of the 'head office' participants. We will see what we can do. In the meantime, why not write a learned article for HC on your application?)

Lisburn

Hint for Hard Core: on ITT 2020, with Pascal 1.1. Shift M does not give a right square bracket]!.
With complements
I P N McKelvie

(There was a column started on such tips -where are all the other ideas.)

Budleigh Salterton

Dear Sirs,
I have recently become aware of the existence of your group, having spoken to one of your members in Leicester, Mr R. C Henson....Perhaps you could let me know if there is a sub-group of BASUG in the south-west; if there is not, I should be prepared to assist in setting one up.

Yours faithfully,

Robert A Mason

(Ed. We are gradually getting our ideas organised on the local group scene. There are members in your part of the world but as yet no group. We will be letting you know and passing on to you any contacts that your letter generates.)

Corbridge

Dear Sir,

Could I make the following points:- i) Would it be possible to include a report of the meetings, such as the one scheduled for January 17th on "Communications", since I am certain that many members are interested in such topics but are prevented from attending by the distances involved. ii) Could you include an evaluation of color cards and monitors in the near future, both UHF and RGB output, if possible. iii) Is there any demand from members for a BASUG group in the northeast centred on Newcastle. I would be willing to help set up such a group.

Yours faithfully

David Steward.

(Tony Williams replies: it is my job to report on meetings for your benefit. I do what I can but in three hours many words are spoken, so my report can only touch on major points, at best. Sometimes. I hope my report in this Hard Core on the communications meetings met some of your needs. Wearing my Membership Sec hat, I can say, off the top of it, that oddly enough there are not a vast number of BASUG members in the northeast as yet. But they are growing, and you can help. See my answer to the member in Devon).

Leamington Spa

I would be willing to act as a focus for a local branch (including Kenilworth, Warwick, possibly Coventry, Rugby, Stratford). Thanks for your efforts.

A Graham-Bryce

(Ed. Thanks for yours. We will pass on to you all inquiries.)

Northwood

Dear John

... You might like to know that the initial response of people to our idea about a new journal on the laboratory use of microcomputers has been exceptionally good even after such a short time.

With best wishes.

P.J.Farago

(Thanks for advertising, Peter. There are a number of members using them in the laboratory, and I particularly, would like to hear what is going on, being an analytical chemist myself... John Sharp)

Farnham

[illegible]

Dear Tony,

I thought the members in our area would like to know that a few computer freaks in the Farnham and Aldershot area have formed a club which is meeting monthly on the second Wednesday at Aldershot library. Our last meeting attracted 31 people and we would welcome new members. Further to this some of the club are starting a ComputerTown UK at Farnham College, Morley Road on Saturday mornings, 9-12. Anybody who is interested in either venture should ring me at the above number for further information.

Regards,

Peter Wicks

Great Missenden

Dear John,

I have acquired a copy of Apple Writer which runs successfully on a Centronics 737 printer. I want, however, to run it on an Epson MX82, but I find that it prints everything on the same line (the Epson has the graphics card). Is there a routine which I could have which would make the Apple Writer operate on the Epson. ... I have only just entered the Apple field and would welcome any courses in the London region - so I am following the BASUG with interest.

Yours sincerely,

Alastair Watson-Gandy

PS. A useful tip, which was passed on to me. I could not get the backward bracket] using Shift M in Pascal. It works perfectly in BASIC. The secret, if this happens is to use Control,Shift M. The advice saved me hours of endeavor, it could save some else too.

(You seem to have a problem, either with your printer, or with not setting APPLEWRITER up properly; you must set the data bit in the Print Constants table. I have an EPSON and use it all the time. John Sharp)

Farnborough

Herewith much thanks for the enormous help given by the Group to lone Apple Users in the sticks, such as myself.

Good wishes for 1981
R A Fairthorne.

(We would like to thank Robert and all the rest of the members who have made similar comments when sending their renewal memberships in. We are getting a tremendous kick out doing it otherwise we wouldn't be doing it. BUT we rely on your support so keep supplying the information and articles.)

Pontefract

Dear Sir,

... I run an ITT 2020 48k with DOS 3.2.1. I am also a radio amateur, call sign G6 BGN and would like to hear of other members who use their machines in the radio field.
I would like to help in the copying and distribution of software from the library.

Yours faithfully

A Lightowler

(Yes, we have ways of making you talk to our members - over the air. Look elsewhere in this edition. We would like to ask someone to spend a bit of time running a SIG for RADIO BUFFS. But we can't do everything; someone else has to put the effort in as well. Sorry to put you on the spot, but I don't recall a reply to my letter suggesting you take this on.
On the copying side, we have immense troubles with copying and distribution even when the members doing copying are so close, so with your being so far away, it is just out of the question. Thanks for your offer, but the admin problems would just not work.)

Herstmonceux

Dear Sir,

...I found the article in Vol. 5 "Enhanced Mode with Epson" very useful. I changed the programme to "Printer" and it works perfectly with "Magic Window" and loads automatically with RAM Printer Driver option selected.
Also found I could use continuous double width using it with a single command from within Magic Window - whereas condensed mode reverts after a carriage return.

Capt. Hanchard-Goodwin

(Anyone else found it useful with other word processors?)

Polytechnic of North London
Dept of Accounting
Holloway
London N7 8DB

Dear Mr Sharp

I am writing to you to enlist your help in connection with the research project that I am undertaking at City University entitled "Application of Microcomputers to small businesses". I would very much like to get in touch with any small business users in your group, to discuss what use they have made of micros and what sort of problems they have encountered.
I emphasise that I myself have considerable experienc of commercial computing, and also of micros and would be happy to give advice to pepole in return for their help on my project.
I should be very grateful for anything that you could so to put me in touch with relevant users in the London area.

Yours sincerely,

Frank Blewitt,
Principal Lecturer in Computing.

Stratford
London

Dear Mr Bolton,
There must be many proud owners of Epson printers with the new Epson type 2 Apple interface who are baffled by the manual's errors, omissions and ambiguities, and by the strange behaviour of the interface itself (e.g. unwanted blanks appearing in Applewriter files.)
I have spent some time attempting to decode the interface ROM programs. Readers wishing to share this admittedly incomplete information and discover some "fixes" are invited to send an s.a.e. to me at [redacted] along with the results of any investigations they may have made. The ROM contents can be printed with the following instructions: PR#1 CALL -151 C800LLLLLL....
Your sincerely,
Robert D. Purves

ITT HIRES PICTURES AND HOW TO SAVE THEM

By Les Budgen.

Most users of the ITT probably already have the ITT save/load a picture program and the associated machine code routines.
The method given below has the advantage of not requiring a load program. This means that you may load a picture from within a Palsoft program. Its disadvantage is that more disk space is required to store the picture.
The pictures are saved as two binary files one for the ninth bits and one for bits 0 to 7.
A short machine code program given below is required to save your pictures. Enter the program as shown:-

CALL-151 to enter the monitor then type in the following:-

```
300:A9 20 85 27 A9 00 85 26 A2 3F A0 00 B1 26
AD 63 C0 30 06 A9 00 91 26 F0 04 A9 80 91 26 88
D0 EC E6 27 CA D0 E5 60
```

To list the program type 300L

The listing should be as below.

```
0300-  A9 20       LDA  #$20
0302-  85 27       STA  $27
0304-  A9 00       LDA  #$00
0306-  85 26       STA  $26
0308-  A2 3F       LDX  #$3F
030A-  A0 00       LDY  #$00
030C-  B1 26       LDA  ($26),Y
030E-  AD 63 C0    LDA  $C063
0311-  30 06       BMI  $0319
0313-  A9 00       LDA  #$00
0315-  91 26       STA  ($26),Y
0317-  F0 04       BEQ  $031D
0319-  A9 80       LDA  #$80
031B-  91 26       STA  ($26),Y
031D-  88          DEY
031E-  D0 EC       BNE  $030C
0320-  E6 27       INC  $27
0322-  CA          DEX
0323-  D0 E5       BNE  $030A
0325-  60          RTS
```

Return to BASIC....3D0G .
3SAVE ITT BIT 9,A$300,L$26

To save your pictures proceed as follows:-
1.BLOAD ITT BIT 9
2.Generate your picture(s).
3.BSAVE (name),A$2000,L$2000 for page 1
4.BSAVE (name),A$4000,L$2000 for page 2
5.CALL 768
6.BSAVE (name)-9 as step 3 and/or 4

You now have two binary files per picture one (name) and one (name)-9.

To load a picture:-
1.POKE 49246,0
2.BLOAD (name)-9
3.POKE 49247,0
4.BLOAD (name)

You may load a picture in this manner in immediate mode or from within a program. From within a program the DOS commands must be prefixed with CHR$(4), see your DOS book.
If you wish to see the picture being loaded enter graphics mode first with HGR or HGR2 before 1.above.

For those who wish to know what happens read on :-
The 360 dot wide graphics of the ITT is achieved by using all 8 bits in each of 40 bytes. Plus 1 bit extra for each byte (often called the ninth bit). Note that the Apple uses only 7 bits of each of the 40 bytes. The eighth is used to select colour. This of course gives a graphics screen 280 dots wide.
A dot on the screen is on or off according to whether there is a 1 or a 0 in the corresponding bit of the appropriate byte. That's easy for bits 0 to 7 (the first 8 bits). Bit 9, however, is somewhat more complicated. As we cannot have 9 bits in a byte, another way had to be found to enter data into bit 9 and to get that data out again.
To enter data a latch (switch) at location Hex C05E is enabled, the data a 1 or a 0 is then put into bit 7 of the byte and because of the latch bit 9 takes this value also. The latch is then disabled by referencing location Hex C05F thus locking in bit 9. Bit 7 is then set to the desired 1 or 0. Palsoft does all this automatically when HPLOTING.
Getting the data out is achieved as follows...When a particular byte within the hires screen is referenced e.g. by PEEKING then the 1 or 0 of the corresponding bit 9 is loaded into bit 7 of a register at location Hex C063, this location may then be examined to see if bit 7 is on or off.
e.g. PRINT PEEK(49251)>127
The listed machine code program references every screen byte, collects bit 9 in each case from C063 and deposits it in the corresponding bit 7 of the screen byte. Bits 0 to 6 are set to zero.
The second binary file that we saved is a file of these bit sevens that will, when loaded, become bit nines. The first is a file of bits 0 to 7.
When we load the picture! - The POKE 49246,0 (C05E) enables the latch mentioned before. Step 2 loads the file of bit sevens which, because the latch is enabled are also entered into the bit nines. POKE 49247,0 (C05F) disables the latch thus locking in the bit nines. Step 4 then loads bits 0 to 7, which of course replaces the temporary values of the bit sevens used in loading the bit nines.

This all sounds complicated but if you follow it through carefully it is in fact quite simple.

apple speller for apple writer etc.,etc.

THE APPLE SPELLER fills the void that has consistently kept the large variety of excellent word processing packages for the Apple II Computer from approaching the power of a dedicated work processor. Finally, the first professional quality spelling verification program is available for the Apple II. The Apple Speller will certainly be the standard against which all other similar programs are compared.

The Apple Speller interfaces to the most popular Apple word processors, including Applewriter, Apple Pie, Executive Secretary, Letter Perfect, Magic Window, and Superscribe II, just to name a few. In fact, The Apple Speller can analyze the output of any editor that writes a standard Apple binary or text file to a diskette. In addition to this flexibility, the performance of The Apple Speller will astound the microcomputer world.

The Apple Speller is supplied with a 30,000+ word dictionary on a single 5¼" diskette with additional space to easily add another 8,000 words to suit your individual needs.

The Apple Speller has built-in utilities to maintain the dictionary diskette. You can readily add words, delete words, and create an unlimited number of modified and/or new dictionaries for specific application.

The Apple Speller is unbelievably fast. The first pass reads your document and collects all the words it contains at a rate of 5,000 words per minute. Next, the words are compared to the dictionary for spelling errors at the incredible speed of 50,000 words per minute. Finally, all misspelled words are marked as such in your document with a rate of 1,000 words per minute. This translates to proofreading a 10 page document in 1 minute if there are no spelling mistakes and 2 minutes, 15 seconds for an unlimited number of spelling errors.

Numerous options are provided throughout the program to enable you to completely control all activities of The Apple Speller. These include the ability to ignore both control codes and formatting commands, an alphabetical listing of either misspelled words or all the words in your document along with usage frequencies, multiple options for the action taken with each misspelled word, and much, much more! A verification mode is provided to allow you to examine and dispense with misspelled words while viewing them in the actual context in which they appeared in your file.

The Apple Speller requires an Apple II/Apple II+ equipped with 48K, DOS 3.3, and 1 or 2 disk drives. Two disk drives are required to delete or add words to the dictionary.

introductory price!

£48-50 INCL VAT & P&P

MICROSOURCE

1 Branch Road, Park Street, St. Albans.
Tel: Park Street (0727) 72917

applefan™ by hiteck

Are you overheating?

Cool things down with the APPLEFAN!
Your Apple will not only run more efficiently, with less risk of breakdown, but will be well organised, with easy access to all input/output ports.

Simple to fit, no soldering or drilling – the APPLEFAN still allows access to the lid of your Apple.
APPLEFAN may be removed at any time.

Basic model only **£68.60** +VAT.

Ask your Apple dealer for a demonstration, or post the coupon for our brochure!

Hiteck Products 21 Station Rd, Knebworth, Herts. Tel 0438 812137

Please send me your brochure of APPLEFAN!

NAME: _____

ADDRESS: _____

_____

_____

MAGIC WINDOW WORD PROCESSOR

Here are two reviews of the 'Magic Window' word-processor, which is available from S.B.D.Software, who kindly lent a copy for review.

REVIEW:THE MAGIC WINDOW

By Tony Williams

In the past months the eyes of the nation have been riveted on the picture appearing in the computer press of a curiously winged and benighted belly-dancer in a tiara casually tossing a stick of dynamite behind her. I can now reveal that this lady is Susan Ben David - she of SBD Software fame - who has kindly lent Hard Core a copy of The Magic Window for review. This, proclaims the text over Susan's left wing, is a "Dynamic Revolutionary Concept in Word Processing", which at one brush of a magic wand can transform your 40 into an 80-column screen without hardware modifications. Well, it doesn't really do that, of course, but by scrolling in four directions and putting up lines representing page edges on screen it kids you into believing that you are seeing an actual page on screen.

This WP package published by ARTSCII INC. of Burbank Blvd, Hollywood, and "created" by Gary Shannon, is retailed in this country at £79.95 + VAT. The package was revised and documented by Bill Depew, who must also be held responsible for one of the package's more abyssmal presentation deficiencies. Programmers who conceive and implement WP software should also show in their documentation that they understand what word processing is for. Namely, by releasing you from the chore of applying white paint to paper and constant retyping with carbon copies, etc. WP frees more time to concentrate on getting the text just right. This includes writing style and also spelling. The documentation for the Magic Window is so full of typos and gross spelling errors that I feel justified in asking whether the authors are fully conversant with what WP is all about and by extension to question whether they have been sloppy in their programming as well. One can forgive stylistic and other peccadilloes in other fields of computing, but not WP.

In all other respects the presentation is quite pleasing. The one disk (no backup facility available) comes in a brown padded ring binder, and includes a large coloured card showing the various key and control commands. This is rather too large to prop up in front of you while working, but the authors have also thoughtfully provided a smaller version which can be removed from the binder. Happily the manual runs to only 55 pages, an indication that mastering the system is not going to take as long as learning to fly a Boeing 747. The system has some nice features. After booting up in the normal way, you are presented with a menu of the following options:

EDITOR SUBSYSTEM, FORMAT SUBSYSTEM, FILER SUBSYSTEM, PRINTER SUBSYSTEM, CONFIGURATION, and EXIT TO BASIC.

You enter each subsystem very quickly and efficiently by moving a bar of light to the appropriate place using the forward and reverse arrow keys. An attractive feature. By entering the format subsystem first you can configure the 'page' as it will ultimately appear on the screen Another winning feature: this configuration can, if you wish, be stored away on disk, together with the file for printing later. With Apple Writer, by contrast, the print constants refer only to the configuration of the document last worked on and have to be revamped for each printing. The Magic Window has a built-in provision for handling various lower case adapters for the Apple such as Dan Paymar's and indeed the manual makes the sensible observation that these devices are so cheap nowadays that no one should hesitate in acquiring one. (Members of BASUG have better options than Dan Paymar's, however). Upper case is accessed by a single touch of the ESC key and locked in with a double stroke. If no adapter is installed the upper case characters are represented by inverse video. As it happens the shift key modification which I use is inoperative, although Dave Bolton has promised to make it compatible with The Magic Window as well as Apple Writer.

This package makes use of a complex battery of keyboard commands. I intend to give only a brief overview here of the features that drew my attention. Right and left cursor movement is activated by the left and right arrows, which in fact scroll the whole body of text past a fixed 'cursor' in the centre of the screen. This is all right if you are not disturbed by the optical spectacle of slabs of text on the move. The up and down movement is effected by CTRL Q (up) and CTRL Z (down) - not such an obvious idea. The letters of the keys are not exploited to give a mnemonic clue to their control function, as was attempted with Apple Writer. CTRL E, for instance, means move to the beginning not the end of the text. CTRL X - quite extraordinary this - means move down half a screen. To insert a single word or body of text you use a strange feature known as Split (Control T) and Glue (Control G). This works quite effectively but seems unnecessarily complicated compared with Apple Writer's ultra simple method of inserting text. Much of the process of editing and relocating chunks of text using Magic Window in fact bears an uncanny resemblance to the physical process of wielding scissors and paste. You stick a bit in here, you have to make room for it by carving a hole - and you also have to close up the hole it came from. If you divide a word to insert a letter (using the Split function) and then Glue it together again it disconcertingly inserts an unwanted space. Where Magic Window gains is that whenever its cursor 'moves' it does not have the disturbing effect of displacing the text by one character. You insert a character at its exact position, not one along. This is useful.

Line justification is actuated not in the format constants but inside Editor Mode, to enable you to visualise the final effect more clearly. This indeed is the central core of the Magic Window philosophy. For the most part this is successful, with some reservations: if fill justify is attempted with a Centronics 737 it appears to work on screen but with the proportional character set selected the right edge of the text emerges ragged on paper. (This also applied to Apple Writer until Ian Trackman fixed it with his Go-Between). CTRL J puts you into Justify Mode (monitored by a status line at the top of the screen). In this mode hitting the key C (not CTRL C) gives Centre Justify, the key L gives Left Justify and R Right Justify. The key E strangely enough means Expand - i.e. left and right justify, what is normally called Fill Justify. The expansion is effected by insertion of spaces. Once in Justify Mode the function is carried out line by line, and the writer is in total control. He can whizz through the text justifying the lot by using the Repeat key, but beware - the Fill Justify mode will absurdly pack spaces into lines that may contain just two words, creating a bizarre effect needing subsequent correction with the Pack line function (Key P). This removes spaces and left justifies. All this is much more complicated than Apple Writer but much more sophisticated in the sense that it enables the writer to work through the text line by line systematically ironing out anomalies on

screen rather than being rudely surprised by the final printed version. Anything that saves us the chore of printing out yet another a draft before the finally acceptable version is to be greeted. Magic Window does not need or support artificial 'wrap around' for words at the end of a line, so the user is not faced with that esthetic monstrosity. However, I have yet to discover just how the advertized 'spellbinder' the Webster's Word-Speller/Divider works or even how to call it up. No reference to it could be found in the manual.

The question still to be answered is, who needs it? Do these extra facilities really justify the (rather modest) extra price vis-a-vis Apple Writer? The answer must lie with the user's personal preference. Some Apple users are known to be disenchanted with Apple Writer's screen conventions and its rather restricted albeit robust facilities. Let's face it, Apple Writer can really drag its feet in pulling its various print options from file and registering changes, etc. By comparison MW is a lovely little mover. In looking round for low cost alternatives they should certainly take a look at the Magic Window. This might not be quite the software for the creative writer who needs to see the beginning and end of the sentence he is currently working on and who is capable of envisaging in his mind's eye the final printed document. However, there is one trick for you to play with. If you do insist on seeing the whole body of text on screen at once, you can format the screen down to 40 characters. This facilitates entering passages of text, checking for meaning, etc. Then, when the creative side is ready, you can reconfigure it to 50 or however many characters you intend it to be and get down to page formatting. A little fiddly, to be sure, but it goes some way toward overcoming such objections. However, in many ways this compromise (for like Apple Writer it is also a compromise attempt to overcome the Apple's 40 column video restriction without going to the considerable expense of an 80-Column card) is likely to please a greater number of users - namely those who want to get their format correct before their eyes. Magic Window lacks one feature present in Apple Writer and leading me again to surmise that the creators know more about programming than they do about user needs. On Page 36 of the manual in describing the configuration of the control keys they refer to "seldom used search commands" placed in the middle of the keyboard. Seldom used? I use them constantly. I have sleepless nights at the prospect of being denied automatic Global Search and Replace. The MW does indeed have a global string search facility which is rather easier to grasp than Apple Writer's (despite its oddball feature of beginning the search by returning to the jumping off point instead of going straight to the string). In its favour I liked the fact that you could find a string, change it, and then continue with the search (by keying CTRL N). Nevertheless I do insist on my automatic replace facility. What would happen if I wanted to change all my Apples to Pears? (Whoops!).

This has been of necessity a brief overview, designed to give a first feel for this package rather than a thorough and meticulous investigation. There are many attractive features I have not been able to get round to. I ought also add that when the much heralded Mailer add-on finally makes its appearance the Magic Window will gain enormously in its power and attraction for potential users.

The article you are reading was written using The Magic Window. Despite what I have written, it still took me four draft printouts before I was satisfied. Since Fill Justify does not operate with the Centronics proportional character set (the Hard Core norm), the standard characters have been selected and reduced photographically.

MASS MICROS

TEL WELWYN GARDEN (07073) 31436

WELLSON HOUSE, BROWNFIELDS
WELWYN GARDEN CITY AL7 1AR

MAGIC WINDOW
A BEGINNER'S EYE VIEW

By Chris Jones

I've had my Apple, a Centronics printer and a Magic Window word processing system for about a month now, and have been using them for writing educational materials. Coming to the Magic Window as a word-processing novice, I found the system fairly easy to use, and confess that I haven't touched my typewriter since.

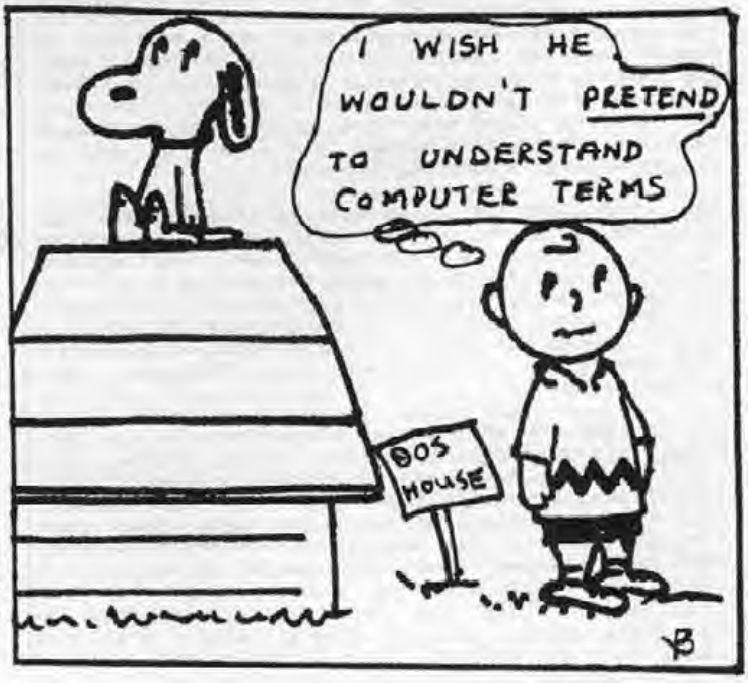
The system has a wide range of editing functions, including tabs, justification, a search function, and ways of moving lines around within a text file and from one file to another. The choice of keys for editing commands seems sensible, with the most common command keys grouped near the Control key for easy one-hand operation, and a colour command chart is provided.

The Magic Window cursor remains in the centre of the screen, while the text scrolls from right to left as you type. This allows you to format 80-column text on a 40-column screen as you go along, though this does have the slight disadvantage that you can't look over what you've written without either scrolling back and forth to read each line, or printing out the page. This hasn't bothered me particularly so far, but it would be nice if a 'checking' function were included which would re-format a file into 40 columns for you to read it through quickly, and then put you back in your original format. Of course, it is possible to type your file in 40 columns, check it through, and then re-format it into 80 columns later, but this involves a lot of messing about, and the whole point of the Magic Window system is to avoid having to do the typing and formatting as separate operations.

My two major complaints concern the use of control characters. While they don't appear on the print-out, Magic Window control characters <u>do</u> take up space in the text line in which they appear. The two characters used to underline the 'do' above reduced the number of spaces on that line available for text from 40 to 38, as you can see. Not that this matters much if you're underlining occasional words here and there, but it does rather complicate things when you're typing in, say, tables across the page, and you're doing quite a bit of underlining and using some expanded print. If you format your page so that it looks right on the screen, without taking into account all the control characters that you've used, the page will be all over the place. This, again, can be handled once you're used to it, but doesn't quite square with the claim in the MW handbook that "What you see is truly what you get". This limitation must also play havoc with anyone trying to right-justify a text using dot-space control commands.

I came across the other problem when I wanted to use small print for the instructions for an exercise I was writing. I typed in the appropriate characters from the Handbook's command sheet, and nothing happened: I was still in 10 cpi. Worried that I might be doing something wrong, I contacted SBD Software, who were extremely helpful, even to the point of coming to my house to try and sort it out. They, too, came to the conclusion that you can't change the print size from within the Magic Window (which you should be able to do - why else list the right control characters in the command sheet?). They gave me a couple of little programs that will put the printer into Proportional and 16.7 cpi <u>before</u> going into the Magic Window, and said they'd be asking the manufacturers about the problem. I haven't heard anything yet, but then there might be some very simple solution which we've all overlooked.

In conclusion, I must repeat that I like my Magic Window. On the whole I find it simple to use, and would be quite happy to put up with the other difficulties I've mentioned if only I could switch print sizes within a file, which you are obviously intended to be able to do. If that problem were solved, I would consider the Magic Window to be good value at £80 + VAT.



Pete & Pam Computers

apple computer

Competitive Quotes and Personal Service
Ring Chris Gillard in London — Pete or Pam Fisher in Lancashire

APPLE SYSTEM SALES AND SERVICE
IN BOTH LONDON AND LANCASHIRE

Now over 500 items for APPLE in stock

Full catalogue and description are available

All at £12.95

All at £14.95

All at £20.95

All at £29.95

OLYMPIC DECATHLON from Microsoft £12.95

FLIGHT SIMULATOR by Sub Logic Disk £29.95 Cassette £24.95

DRAGON FIRE from Dakin Corp £29.95

VERSA EXPANDER PORT £12.95

SENSIBLE SOFTWARE UTILITIES
Applesoft Programme Optimiser £12.95
Multi Disk III £34.95
Super Disk Copy III £17.95
DOS Plus £17.95
Disk Organiser II £17.95
Disk Recovery £17.95
Applesoft Plus Structured Basic £14.95

DAKIN 5 PROG All 5.3.3 £49.00

PASCAL TUTOR £79.00

PASCAL PROGRAMMER £79.00

RAMCARD £89.00

RAMEX 16 £75.00

MEMORY MANAGEMENT SYSTEM £29.00

Z80 SOFTCARD £195.00

COBOL 80 £299.00
FORTRAN 80 £109.95
BASIC COMPILER £199.00
ASSEMBLY LANGUAGE DEVELOPMENT (6502, 8080 and Z80) £79.00

2 TERM £59.00

80 COLUMN BOARDS
SUP-R-TERMINAL £195.00
VIDEX £185.00

NEC GREEN SCREEN MONITOR £169.00

WORDSTAR for APPLE £169.00

MAILMERGE £69.00

SOFTKEY £74.00

ART NUMERICAL KEYPAD £39.00

VISICALC 3.3 £74.95
Our Price £105.00

VISIDEX £105.00

VISITERM £79.00

VISIPLOT £85.00

VISITREND £129.00

VISIFILE £139.00

DB MASTER £129.95

DB MASTER Utility Pack No. 1 £60.00

DB MASTER for Corvus — available soon £295.00

INFORMATION MASTER — Data Base £79.00

DATA MASTER £55.00

TRANSIT £29.00

UP TO 82K FOR VISICALC

Saturn System 32K Ram board and their software package.

Visicalc-Expand can give additional memory for Visicalc applications.
32K single board gives 47K.
16K card + one Saturn board gives 66K.
2 Saturn boards give 82K.
Board comes with DOS relocation, pseudo disk Applesoft integer utilities.

Single Board £149.00
Visicalc-Expand software £55.00

APPLE OS9
STELLATION TWO'S MILL 6809

Is available with OS9 and BASIC O9 — NOW.
BASIC O9 allows simultaneous running of separate programs. It has to be seen to be believed — (OS9 is modelled along the lines of UNIX).

£399.00 + VAT

PASCAL JOB CONTROL SYSTEM — from High Technology £295.00

REMOTE OPERATING SYSTEM — ROS £499.00

64K RAM CARDS £124.00
£449.00
£225.00

TASC — THE APPLESOFT COMPILER £109.00

THE MILL — A 6809 plug in board for Apple £249.95

APPLE MUSIC THEORY £29.95

MOUNTAIN CPS — Multi-function Card £135.00

VISIWRITER £149.95

VERSA EXPANSION SOFTWARE £20.95

EPSON MX-80 F/T £398.00

EPSON MX-80 T £349.00

INTERFACE AND CABLE £65.00

GRAPPLER from Orange Micro £99.00

AIO INTERFACE from SSM £119.95

MACHINE COVERS — only the best material used
Apple only £5.95
Single Disk £2.95
2 stacked disks £4.45
Apple, 2 disks and 9" monitor or Apple and 12" monitor £8.95
Apple and 2 disk £7.95
Epson MX 70/80 £5.45
Paper Tiger 445 – 460 £5.45

BASF DISKS (for 10) £18.50

Authorised Apple Sales and Service

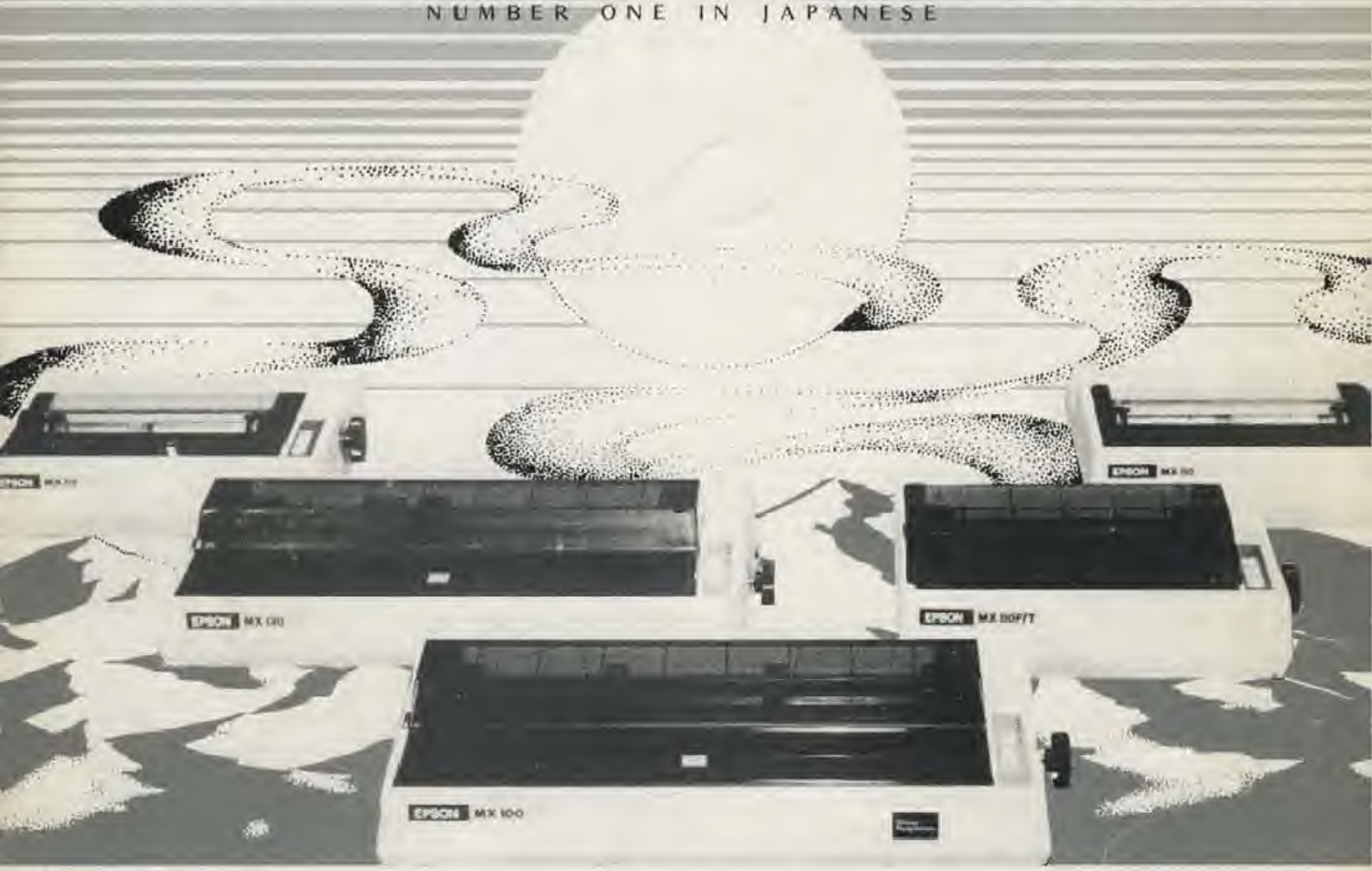
LONDON RETAIL

MAIL ORDER AND DISTRIBUTION

VISA

ICHIBAN

NUMBER ONE IN JAPANESE



Highest possible quality + lowest possible cost = EPSON +MICRO PERIPHERALS
for the best of both worlds.

	Matrix			Paper Handling		Speed		Max. Paper Width		High Res. Graphics & Italics
	7 x 5	9 x 7	9 x 9	Friction	Tractor	80 CPS	135 CPS	10"	15½"	
MX80 T			●		●	●		●		
MX80 FT/1			●	●	●	●		●		
MX80 FT/New Type 2			●	●	●	●		●		●
MX82			●		●	●		●		●
MX100			●	●	●	●			●	●
MX130		●		●	●		●		●	●

The above machines have many more features including interfaces for Apple, PET (with PET Graphics), TRS80, Sharp, NEC, Hitachi, Nacom, Acorn, Super Brain, Video Genie, BBC Micro etc, some have correspondence quality printing and multiple character sets including international languages. Ring Ian today for full details and specifications and printout samples. All machines usually ex-stock with next day delivery plus 12 months no-quibble guarantee.

Wholesalers and Distributors of quality Japanese Micro Products.

Micro Peripherals ltd.

61 NEW MARKET SQUARE, BASINGSTOKE, HANTS. Telephone: 0256 56468

Europe's leading Epson Distributor and Supplier. Trade Sales - 0256 56469

JAPANESE OFFICE: 101 Abe Bldg. 4F, 2-42 Kanda Jinbocho, Chiyado-ku, Tokyo, Japan.